UNIVERSITY OF WEST ATTICA

eISSN:3136-7658

Applied Manufacturing Intelligence Journal

Vol. 5, No. 4, October, 2026



Small-Sample Masked Graph Autoencoding for Collision-Risk Prediction in Underground Mine Locomotive Operations

Maria Kijewska[1,*], Marcel Żak[1]

[1] *Faculty of Mechanical Engineering and Automatics, Rzeszow University of Technology, Rzeszow, 35-959, Poland*

Received 31 May 2026; Revised 24 September 2026; Accepted 30 September 2026; Published 03 October 2026

Abstract

Collision-risk prediction for underground mine locomotives is difficult because verified near-conflict labels are scarce, telemetry is incomplete, and risk depends on interactions among vehicles, track sections, and switches. This study evaluates a small-sample Masked Graph Autoencoder (MGAE) that reconstructs masked node attributes and operational edge context before supervised risk prediction. A haulage-network dataset comprising 4,860 operating windows, 37 graph nodes, and 52 directed section edges was used. The evaluation considered labeled fractions of 5%, 10%, 20%, and 40%, with episode-aware partitioning and matched normal windows. At the 20% labeled condition, the proposed model attained an F1-score of 0.861, an area under the precision–recall curve of 0.884, balanced accuracy of 0.902, and a reported false-alarm rate of 7.3%. Its mean warning lead time was 18.9 s under the study's event-boundary definition. At 5% labeled data, its F1-score was 0.742, compared with 0.628 for graph attention and 0.584 for the recurrent baseline. Removing edge masking reduced F1 from 0.861 to 0.802. These results indicate that topology-aware reconstruction is promising for the evaluated network and label protocol, while the single-network scope, limited number of independent risk events, and incomplete operational exposure measures restrict generalization. Multi-site, repeated-run, and end-to-end validation are required before deployment claims can be made.

Keywords: Underground Locomotive, Collision-Risk Prediction, Masked Graph Autoencoder, Topology-Aware Masking, Sparse Event Labels

1. Introduction

Research on underground locomotive collision warning shows that the causes of operational risk include route conflicts, reduced visibility, irregular loading cycles, and communication latency [1]. Research on mine haulage networks has shown that narrow drifts, all kinds of track arrangements and switch geometry can affect the safety implications of vehicle speed and separation [2]. Therefore, the predictive warning method will be used to add positioning data, switch-change status, dispatch commands and traction signal information for conflict detection in advance [3]. Other research has also shown that the absence of incident labels and missing underground observations limit the accuracy of supervised risk prediction [4]. At the time of high-production runs, some locomotives needed to use the passing loop and loading approach simultaneously to avoid stopping the main haulage. A safe following move and an opposing approach can be at the same distance and speed, but they will need different dispatch responses. Fixed thresholds cannot easily achieve this division without considering the direction of travel and reservation compatibility. Research on underground sensing has further documented asynchronous sampling and temporary communication interruptions that make synchronization of vehicle and infrastructure states difficult [5]. Therefore, representation learning can be conducted on ordinary operating records to learn from regular movements and recurring interactions,

* Corresponding author: maria.kij@prz.edu.pl

https://doi.org/10.68406/amij.2026.vol5iss4np1:1-13

➢ This is an open access article under the CC BY-NC-SA 4.0 license (https://creativecommons.org/licenses/by/4.0/).

© Authors retain all copyrights.

even in the absence of annotated high-risk episodes. Sparse-label safety prediction methods offer a way to use these records and reduce the cost of event verification [6]. The actual problem is how to preserve the relationships that cause a conflict without creating a model that is too complex for the information available in the labelled sample.

A developing collision can be viewed as the outcome of pre-programmed motion rather than an unforeseen observation by an individual sensor. Therefore, route-conflict models consider shared sections, reservation overlap, switch confirmation and the time of vehicle arrival simultaneously [7]. Graph representation retains these relationships through entities and directed connections, and the importance of a locomotive's state can vary according to its position in the haulage network [8]. A stationary vehicle in the loading area is safe to park but obstructs the active road; therefore, it is not a speed problem. The two locomotives of adjacent branches can be compatible in terms of reservation until they reach a common switch. If the physical proximity, occupancy status, reservation status and proximity relationship all need to be taken into account separately, a graph can be employed. Supervised graph models are unstable in the case of a small number of positive examples and local repetition [9]. A few verified episodes may show only one loading method or a single switch setting; therefore, the model links that particular location to a high-risk area. Therefore, the issue is not only class imbalance but also a lack of labels for certain interaction types. Increasing the Model's capacity will not solve the coverage problem. A good small-sample design should learn the general compatibility of routes and vehicle states before applying a risk interpretation to the relatively few annotated events.

Self-supervised masked graph learning can be used to discover these operating relationships without the need for labels in every window. Masked reconstruction methods hide some attributes and learn representations to recover the missing information from the remaining graph [10]. Periodically, the standard records in the underground haulage environment contain data on feasible route sequences, section occupancy and switch transitions and speed envelopes [11]. Transfer learning of an encoder that has learned these regularities can improve the performance of a downstream predictor with a relatively small amount of labeled data [12]. However, the reconstruction task should be based on the engineering problem. A general speed value is less informative than a reservation conflict in the switch state and nearby traffic. Therefore, the edge attributes should be processed separately and not included in the general adjacency matrix. Mask sampling is also the same; uniformly hidden attributes will cause the training to proceed too linearly and show little interaction at the switches and passing loops. A short-horizon predictor is fast to warn but less accurate in reconstruction. Too close to the normal operating condition, a reconstruction will not show the small defect and thus will not trigger an alarm. Therefore, the joint evaluation of label efficiency, calibration, threshold sensitivity and warning time will be conducted separately, and the improvement of one metric will not be overshadowed by a decline in others.

This study evaluates whether masked reconstruction of operational graph structure can support collision-risk prediction when only a small labeled subset is available. The method combines a heterogeneous graph of locomotives, track sections, and switches with topology-aware masking, edge-context reconstruction, and a calibrated risk head. The evaluation compares label fractions, graph-learning baselines, component ablations, scenario-level errors, and inference cost. The paper is organized as follows: Section 2 reviews underground collision warning and graph representation learning; Section 3 describes graph construction, masking, and prediction; Section 4 reports the dataset, evaluation protocol, comparative results, ablations, and operational analyses; Section 5 states the findings, limitations, and future work.

2. Related Work

2.1. Collision Warning for Underground Mine Locomotives

Traditionally, the collision warning for underground haulage has been achieved by proximity detection, route locking and rule-based dispatch constraints. Radio-frequency tags, ultra-wideband positioning, inertial localization and leakage communication systems have been used to estimate the position of vehicles in areas without satellite positioning [13]. These systems provide sensing support for alarms but generally lack full-function decision-making capabilities; therefore, when two vehicles are close to each other or need to be in different lanes, no alarm is issued [14]. It has a simple logic for the operator and is easy to understand, but it cannot distinguish between an innocent adjacent condition and a serious hazard near the switch. Given the above problems, a model-based approach has been introduced to add braking distance, speed profile, track gradient and signal delay to the warning index [15]. The braking response characteristics of underground mines are affected by load, rail contamination, and driver behavior; therefore, some areas have been provided with a larger fixed safety margin, and others are too small [16]. Data-driven methods, such as support vector machines, random forests and recurrent neural networks, have been employed in the risk recognition of industrial vehicles with labelled data [17]. Their performance generally decreases in the presence of severe class imbalance because the risky state is rare and normal operation has many visually similar, safe trajectories [18].

Another defect of traditional collision prediction is that track topology is generally represented by hand-crafted attributes. A locomotive near a switch is not inherently dangerous; the risk arises from route reservation, opposing traffic, the condition of the switch, and the sequence of connected sections. The graph can present these connections as

interactive components by using a track component and vehicles, rather than individual data items. As most of the conflicts occur at the bottlenecks, branches, loading/unloading areas, and passing loops of the mine transport, this view is especially suitable. Graph Construction should also consider operational semantics. Edges need to store information such as direction, reservation availability and recent occupancy, in addition to being connected. Without the above meanings, a graph model will learn a geometric neighborhood that is weakly related to safety. The existing research has given little attention to the cost of label acquisition. Near-miss labels are generally not in the sensor data, so they need to be linked with position traces, dispatcher decisions, operator reports and section occupancy logs. The speed of this process is low, and there may be different understandings of the same event by production and safety staff. Therefore, a way that needs hundreds or thousands of positive labels is not suitable for many mines. Topology and the operating environment are included in the same graph at the time of risk prediction, and small-sample learning is considered an engineering requirement rather than a secondary evaluation case.

2.2. Graph Representation Learning Under Sparse Labels

Graph neural networks have been widely applied to model relational data in traffic forecasting, equipment interaction and anomaly detection in industrial systems [19]. Message passing can be used to collect information from the neighboring nodes and edges of a node representation in a railway-like network to spread risk along connected sections [20]. For a small-sample problem, direct supervised training may result in an unstable decision boundary due to a lack of positive samples. Self-supervised graph learning addresses this problem by constructing pretext tasks for the unsupervised structure of graphs [21]. Masked graph autoencoders are a representative type of the above idea: some nodes and edges of the graph are hidden, an encoder learns latent representations from the visible graph, and then a decoder reconstructs the missing content [22]. The resulting encoder can also be employed for the downstream task with fewer labels because it has already learned regularities in the graph topology and attributes [23].

Prior masked graph methods provide relevant foundations, but their direct transfer to underground locomotive collision prediction remains an open empirical question. The present task differs in its heterogeneous node types, edge-rich route structure, and short-term risk labels that may change with traffic state. A vehicle-section relation, a switch-route relation, and section adjacency have distinct operational meanings. These considerations motivate reconstructing both node attributes and edge context while preserving route direction, then mapping the learned representation to a calibrated warning score. Calibration is operationally important because dispatchers must balance missed warnings against unnecessary stops or additional verification. The study therefore evaluates a domain-specific masked graph pipeline; it does not assume that a general masked autoencoder is inherently unsuitable or that the proposed design will transfer across mines without validation.

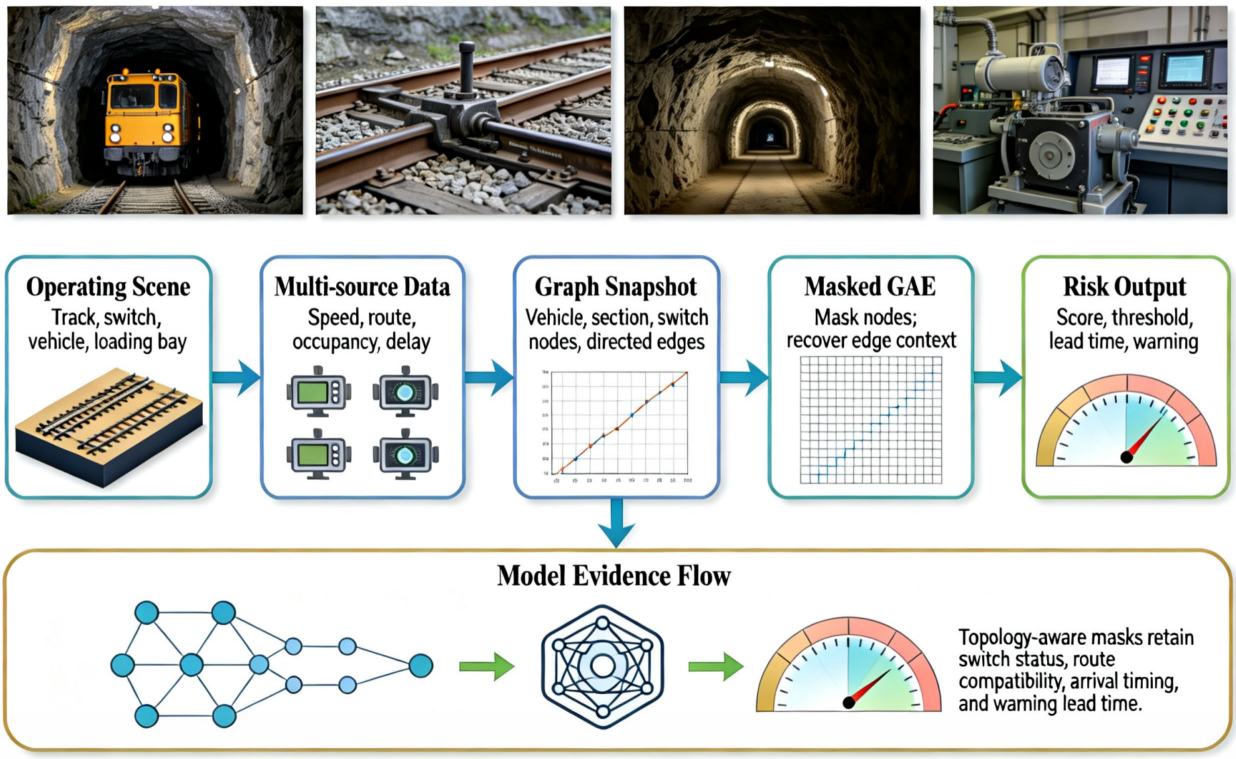


Figure 1. Overall framework of the small-sample masked graph autoencoder for underground locomotive collision risk prediction

3. Proposed Masked Graph Autoencoder Prediction Model

3.1. Graph Construction and Framework

Each operating window is represented as a directed heterogeneous graph. Locomotive nodes contain speed, acceleration, payload class, heading consistency, communication delay, and recent stop duration; section nodes represent track segments; and switch nodes encode route-control locations. Directed edges distinguish physical adjacency, route reservation, and switch connectivity. Node attributes are normalized within each operating day, and the graph snapshot is formed over a five-second window advanced at a two-second stride. The encoder receives node features and relation types, while edge context carries operational information used to distinguish geometrically adjacent but behaviorally different movements. Figure 1 summarizes the data-to-risk pipeline. The available draft does not specify the synchronization or interpolation rule for every telemetry channel; that processing detail should be confirmed from the acquisition implementation before replication.

The masking strategy targets operationally informative context rather than selecting all graph elements uniformly. Node attributes and edge-context fields are masked separately so that the encoder must recover both local state and relations such as route compatibility, switch confirmation, and time-to-arrival difference. The design emphasizes switches, passing loops, and recent occupancy transitions, where local attributes alone may not distinguish a safe wait from a developing route conflict. During pretraining, unlabeled operating windows are used to learn reconstruction representations; labeled near-conflict windows and matched normal windows are reserved for supervised fine-tuning, calibration, and warning-threshold selection. Figure 2 shows this workflow. Exact mask probabilities and their selection procedure should be reported from the training configuration to make the experiment fully reproducible.

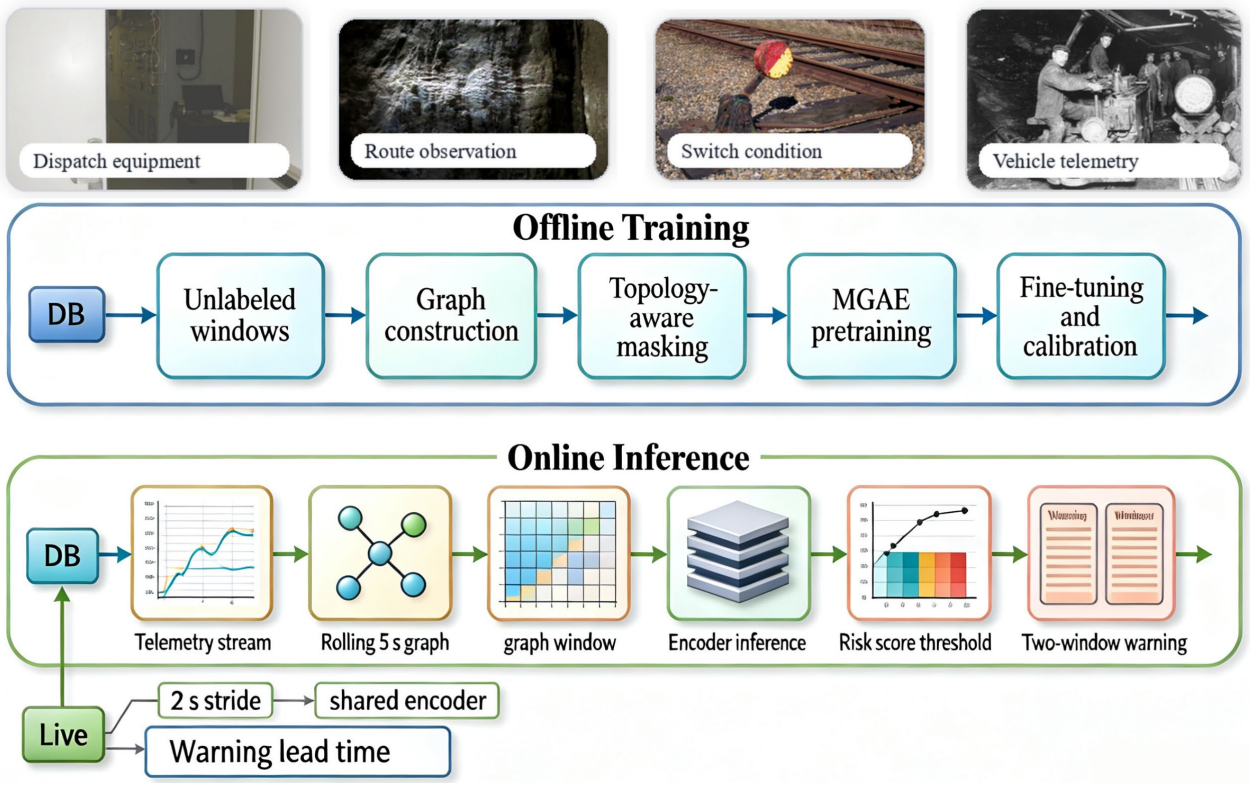


Figure 2. Training and inference workflow from unlabeled graph pretraining to calibrated collision warning

3.2. Masked Reconstruction and Risk Prediction

Let $G_t = (V_t, E_t, X_t, A_t)$ denote the graph snapshot for time window t. V_t contains locomotive, section, and switch nodes; E_t contains directed operational edges; X_t is the node-attribute matrix; and A_t stores edge context. Binary masks M_x and M_e identify hidden node attributes and edge-context entries. The encoder maps the visible graph to latent node embeddings, and the decoder reconstructs the selected components. The reconstruction equation weights node-attribute and edge-context errors separately and evaluates them only over masked entries.

$$L_{rec} = \left\| M_x \odot \left(X_t - \hat{X}_t \right) \right\|_2^2 + \lambda_e \left\| M_e \odot \left(A_t - \hat{A}_t \right) \right\|_2^2$$

The coefficient λ_e controls the relative contribution of edge-context recovery. Relation-aware message passing combines a node representation, neighboring-node information, and operational context on each incoming edge. A

locomotive approaching a shared section can therefore receive different representations according to whether a relation encodes physical connectivity, a route conflict, or a switch-mediated movement. The displayed message-passing equation specifies the encoder update and preserves these typed distinctions within a compact architecture.

$$h_v^{(k+1)} = \sigma\left(W_s h_v^{(k)} + \sum_{u \in N(v)} \alpha_{uv} W_m h_u^{(k)} + \beta_{uv} W_e a_{uv} \right)$$

In the message-passing equation, α_{uv} denotes the normalized relation-attention weight for a message from node u to node v, while β_{uv} scales the associated operational edge context. After K layers, node embeddings are pooled into a graph-level representation and mapped to a short-horizon risk score. This pooled score is calibrated on validation observations using temperature scaling. Threshold selection is tied to the operating trade-off between missed warnings and nuisance alarms, rather than to test labels.

$$r_t = sigmoid(w_r^T Pool(\{h_v^K \mid v \in V_t\}) + b_r)$$

The joint objective combines masked reconstruction, supervised binary cross-entropy, and parameter regularization. Training proceeds in two stages: the encoder and decoder are pretrained on unlabeled graph snapshots for up to 160 epochs, with early stopping based on reconstruction loss; the decoder is then discarded and the encoder is fine-tuned with the risk predictor on the labeled subsets. Windows from the same near-conflict episode are kept in one partition to reduce temporal leakage. At inference, an alarm is confirmed only when the calibrated risk score exceeds the selected threshold for two consecutive windows. Optimizer settings, loss weights, layer dimensions, and random-seed policy should be reported from the experiment configuration.

$$L = L_{rec} + \lambda_c BCE(y_t, r_t) + \lambda_b \|\theta\|_2^2$$

The implementation is designed for a compact underground control workflow. A small number of message-passing layers limits the influence of distant sections on local conflict representations, and relation embeddings are shared across snapshots. Node attributes are normalized within operating days to reduce gradual sensor drift. During pretraining, route-conflict edges receive a higher mask rate than ordinary physical-adjacency edges; during fine-tuning, class weighting is applied to the supervised loss while the reconstruction objective remains unweighted. The exact values for these choices should be included in a reproducibility record.

The encoder retains local operating context by limiting message-passing depth, while shared relation embeddings provide a common representation across graph snapshots. Normalization is performed within operating days to reduce gradual production drift. These implementation choices constrain model capacity and may affect transfer to networks with different topology; the reported aggregate metrics therefore need to be interpreted together with the stated graph schema and evaluation setting.

4. Experiment and Analysis

4.1. Experimental Data and Evaluation Protocol

The case dataset contains 4,860 operating windows associated with a haulage network of 37 graph nodes, 52 directed section edges, 8 switches, 6 passing loops, and 4 loading or unloading stations. Windows with timestamp gaps exceeding 8 s were excluded during preparation. The manuscript dataset includes 126 near-conflict windows identified through route overlap, opposing movement, delayed switch confirmation, and dispatch-log review. Figure 3(a) presents the labeled subsets: the 5%, 10%, 20%, and 40% conditions contain 6, 13, 25, and 50 risky windows alongside 180, 360, 720, and 1,440 matched normal windows. Normal counts are displayed at one tenth of their original scale to keep both classes readable on a common axis. The change in positive coverage is therefore visible without suggesting a balanced training set. Episode-aware partitioning is an established safeguard against leakage from correlated temporal observations [24]. Accordingly, adjacent windows belonging to an annotated episode remain in the same partition. Unlabeled data used for pretraining must likewise come from the training partition if the evaluation is intended to measure inductive deployment performance. Test labels are reserved for the final comparison and do not determine masks, hyperparameters, or thresholds. The small positive counts make this separation particularly consequential, since duplicating even one event across partitions can materially change apparent performance.

Figure 3(b) shows median section occupancy of 0.31, 0.47, and 0.64 in low-, medium-, and high-density operation, with corresponding interquartile intervals of 0.20-0.42, 0.34-0.58, and 0.51-0.76. The intervals overlap substantially; density is therefore a context variable rather than an event label. A high median does not itself establish an incompatible route. Figure 3(c) displays route overlap against arrival-time difference, with reference boundaries at 80 m and 22 s. The concentration of risky examples near long overlap and short arrival difference motivates retaining both spatial and temporal information. These boundaries describe the plotted profile and should not be interpreted as independently

validated safety limits. Matching normal examples by operating context is motivated by research on shortcut learning in imbalanced event recognition [25]. Here, comparison against occupied rather than idle sections make the classification problem more demanding. A predictor must distinguish incompatible approaches from legitimate waiting and passing movements. The scatter also leaves overlap between the classes, indicating why a single distance or timing threshold cannot reproduce the full decision problem. The required evidence includes route direction, reservation status, and switch confirmation, which are represented as separate graph relations.

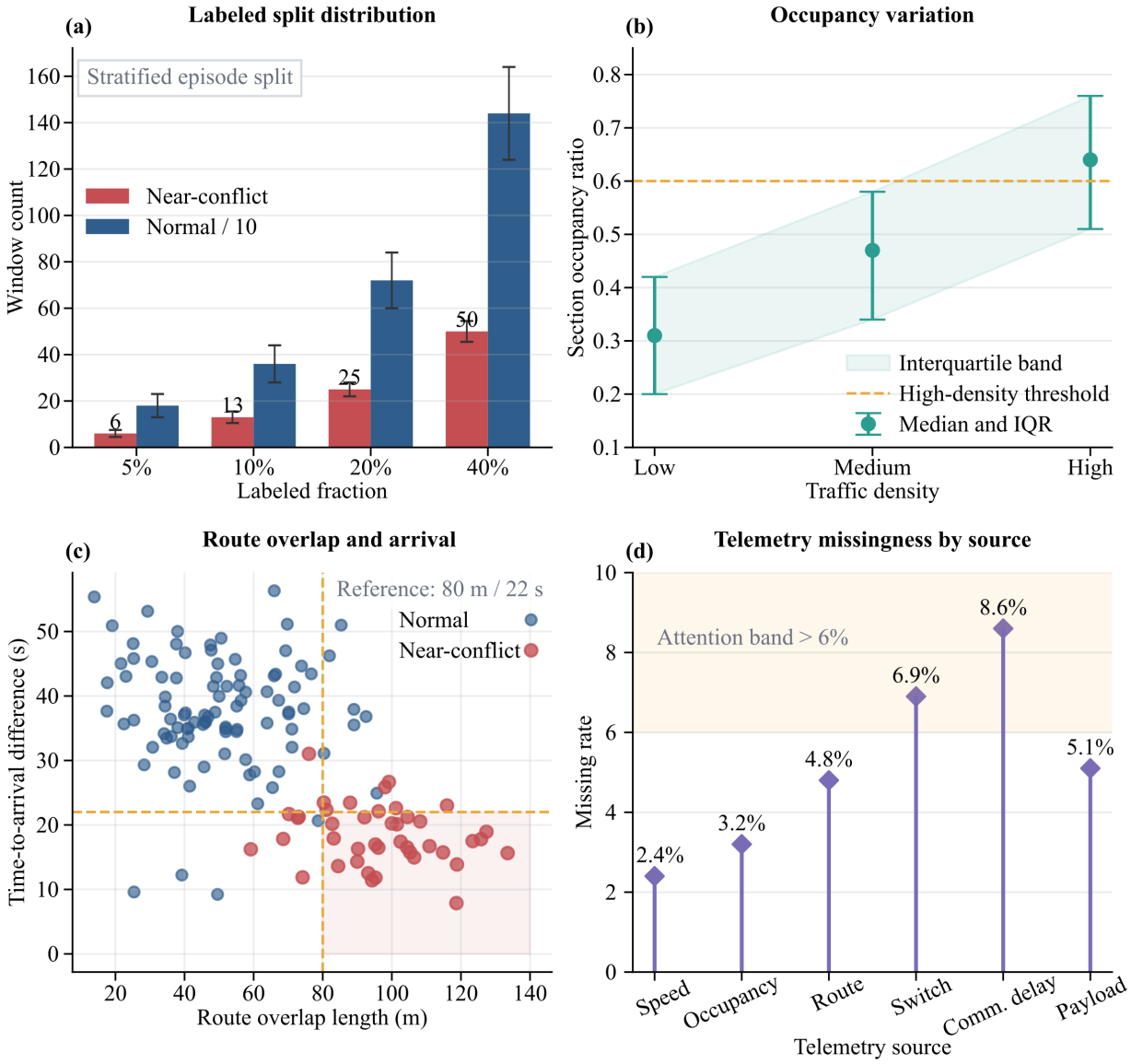


Figure 3. Dataset profile and operating-state characteristics: (a) labeled split distribution; (b) occupancy variation across density levels; (c) route-overlap and time-to-arrival risk map; (d) missingness by telemetry source

Figure 3(d) reports missingness of 2.4% for speed, 3.2% for occupancy, 4.8% for route records, 6.9% for switch fields, 8.6% for communication delay, and 5.1% for payload. The two largest rates affect variables that help interpret whether an apparent conflict is genuine, making missingness relevant to warning reliability rather than merely to data cleaning. Masked reconstruction should therefore be assessed across attribute groups rather than only through an aggregate error. Masking during pretraining and naturally absent telemetry remain distinct: the former creates a learning target, while the latter may have no observed value against which reconstruction can be checked. The evaluation uses F1-score, area under the precision-recall curve, balanced accuracy, episode-level false alarm rate, and warning lead time. Precision-recall evaluation is recommended for rare-event prediction because aggregate accuracy can obscure weak positive-class detection [26]. Baselines comprise logistic regression, random forest, a gated recurrent unit, a temporal convolutional network, a graph convolutional network, and a graph attention network. They use the same available observations and evaluation partitions. Hyperparameter selection is confined to validation data, and preprocessing statistics must be estimated from training observations. This prevents the comparison from attributing preprocessing or split advantages to the graph encoder.

At the 20% labeled condition, Table 1 reports an F1-score of 0.861, AUPRC of 0.884, balanced accuracy of 0.902, a 7.3% false-alarm rate, and a mean lead time of 18.9 s for the proposed model. Relative to graph attention, the reported differences are +0.087 in F1, +0.070 in AUPRC, and +0.035 in balanced accuracy, with a 1.1 percentage-point lower false-alarm rate. Lead time is measured from the first stable warning to the annotated conflict boundary; operational confirmation additionally requires two consecutive windows above threshold. The table presents point estimates, and the manuscript should state the false-alarm denominator and whether results aggregate repeated runs before those values are interpreted as operational rates.

Table 1. Comparative prediction performance under the 20% labeled split

Model	F1-score	AUPRC	Balanced accuracy	False alarm rate	Lead time
Logistic regression	0.593	0.621	0.714	12.8%	9.4 s
Random forest	0.648	0.688	0.762	11.5%	10.7 s
GRU	0.702	0.735	0.811	10.2%	12.1 s
TCN	0.716	0.748	0.824	9.8%	12.7 s
GCN	0.751	0.792	0.851	8.9%	13.4 s
GAT	0.774	0.814	0.867	8.4%	13.6 s
Proposed MGAE	0.861	0.884	0.902	7.3%	18.9 s

4.2. Comparative Performance and Small-Sample Behavior

Figure 4(a) compares the four labeled fractions. Proposed-model F1-scores are 0.742, 0.816, 0.861, and 0.889 at 5%, 10%, 20%, and 40%, respectively. Graph attention reaches 0.628, 0.716, 0.774, and 0.848; graph convolution reaches 0.607, 0.702, 0.751, and 0.823; the recurrent model reaches 0.584, 0.661, 0.702, and 0.769. The proposed-model advantage over graph attention decreases from 0.114 to 0.041 as labeled coverage increases. Thus, the numerical pattern is compatible with a benefit concentrated in the smallest training sets, rather than a constant offset across sample sizes. Self-supervised transfer research provides the rationale for examining performance as a function of labeled coverage [27]. The present curve supplies the manuscript's evidence; it does not establish a general guarantee of sample efficiency. Its largest increment is 0.074 between the 5% and 10% conditions, followed by increments of 0.045 and 0.028. Graph attention gains 0.088, 0.058, and 0.074 over the same intervals, indicating that additional labels can help the supervised model catch up. Repeated episode-level splits and uncertainty intervals would be needed to distinguish a persistent advantage from variation caused by which few positive episodes are selected.

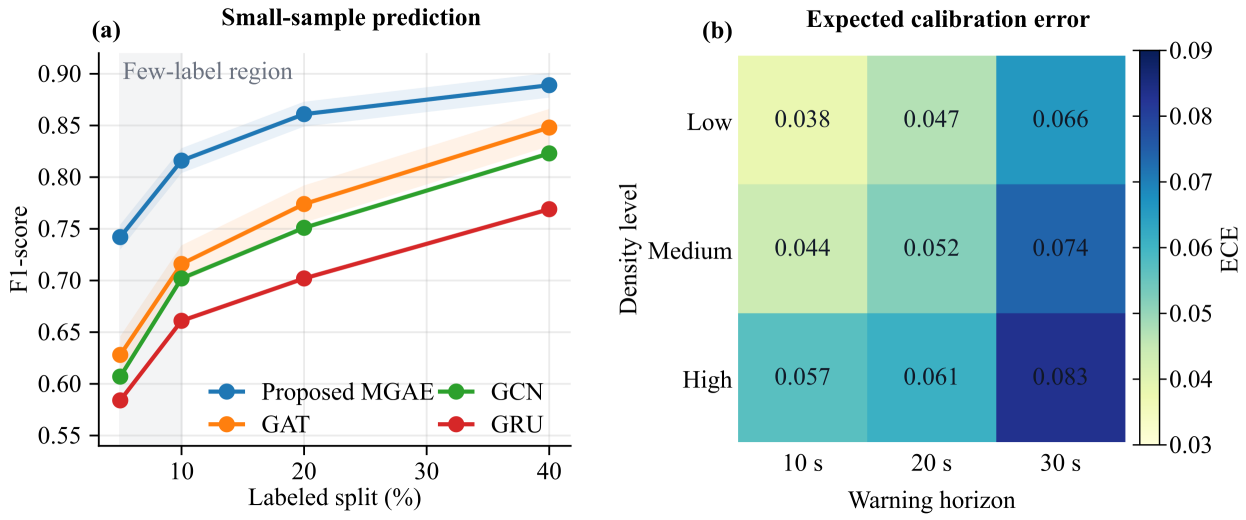


Figure 4. Small-sample comparative behavior: (a) F1-score across labeled splits; (b) calibration error across density levels and warning horizons

Figure 4(b) presents calibration error by density and warning horizon. At 10 s, expected calibration error is 0.038, 0.044, and 0.057 for low, medium, and high density. At 20 s, the corresponding values are 0.047, 0.052, and 0.061; at 30 s, they rise to 0.066, 0.074, and 0.083. Thus, error is higher at longer horizons and, within each horizon, at higher density in the displayed estimates. Gao et al. [28] showed that uncertainty-aware probabilistic graph models can support road-level traffic crash prediction, underscoring the need to distinguish probability reliability from classification ranking.

The highest cell (0.083) identifies a comparatively uncertain condition but does not show whether probabilities are systematically over- or under-confident. Reliability-bin counts and observed frequencies are needed for that diagnosis. Density differences may also reflect event prevalence and subgroup sample size; the figure alone cannot separate these effects. Thresholds should be selected on validation observations, and density-specific adjustment should be considered only when each subgroup has sufficient independent events.

The graph baselines consistently exceed the recurrent baseline at the matched label fractions. Graph convolution exceeds the recurrent model by 0.023 at 5%, 0.041 at 10%, 0.049 at 20%, and 0.054 at 40%. Graph attention adds a further 0.021, 0.014, 0.023, and 0.025 above graph convolution. These comparisons are consistent with useful relational information in the task, but they do not independently measure the effect of route direction or edge semantics. Those factors would require controlled graph-construction ablations. Du et al. [29] used a mixed graph convolutional structure for multi-class trajectory prediction, which motivates preserving interacting entities rather than flattening every observation into an independent feature vector. In the present operating context, parallel-branch traffic and opposing approaches to a common switch may have similar speed sequences while differing in reservation compatibility. A temporal model can learn correlations with that context only if the input encoding exposes it. Conversely, a graph model can still fail when incorrect reservations create misleading edges. The proposed reconstruction stage offers one possible explanation for the observed ranking: it supplies an objective on ordinary traffic before the risk labels are introduced. The performance curve alone cannot prove that explanation, which is why component removal and attribute-level reconstruction behavior are examined separately.

Both the event coverage and the nominal percentage of the low-label condition need to be improved. The six risk windows may not cover all switch geometries, loading modes, and communication failure modes. The value of them varies with different episodes and interaction types, and several neighboring windows of the same event offer little diversity compared to independent events at different locations. The 20% condition has 25 risky windows and a false alarm rate of 7.3%; it is lower than that of graph attention (8.4%) and temporal convolutional networks (9.8%). These are the current evaluation standards, not operators' consents to other mines. They do not report the number of alarms per shift without an exposure denominator. A production line with many minor accidents would still have a relatively high workload at a low episode rate. Threshold review therefore needs event counts, operating durations and the consequences of delayed warnings in addition to the reported indicators. A 40% F1-score of 0.889 shows that more labelled coverage is still beneficial; however, reducing the increment of the proposed model does not significantly increase the sample size. The new labels for the less frequently used loading bay or blind curve may contain more information than the additional labels for frequently used straight-section interactions.

4.3. Ablation and Operational Analysis

Figure 5(a) reports F1-scores of 0.861 for the full model, 0.802 without edge masking, 0.824 without node masking, 0.833 with uniform masking, and 0.847 without calibration. Edge-mask removal produces the largest decrease, 0.059, compared with 0.037 for node-mask removal and 0.028 for uniform sampling. This pattern supports investigating route relationships as a source of predictive information; it does not establish statistical significance without repeated runs. Controlled component ablation is used in representation-learning studies to separate contributions that a baseline comparison can conflate [30]. Figure 5(b) compares lead-time distributions for the full model, the variant without topology-aware masking, and graph attention, with central values near 18.9, 15.8, and 13.6 s. Distribution spread matters because a favorable mean can coexist with short warnings for rapidly developing events. Figure 5(c) displays normal and near-conflict score distributions centered near 0.29 and 0.73, with the warning threshold at 0.62. Their overlapping tails show why separation cannot be reduced to the two centers. Figure 5(d) evaluates thresholds of 0.48, 0.54, 0.62, 0.68, and 0.72: recall decreases from 0.912 to 0.821 while false alarm rate falls from 10.8% to 5.9%. At 0.62, recall is 0.866 and false alarm rate is 7.3%. The selected operating point represents a particular missed-event and alarm-workload trade-off, rather than a universal safety threshold.

Figure 6(a) displays scores before a shared-route conflict. At 20 s before the boundary, proposed-model, graph-attention, and recurrent scores are 0.58, 0.43, and 0.35; at 15 s they are 0.67, 0.54, and 0.48. The proposed score therefore passes 0.62 between these plotted times, before graph attention passes it between 15 s and 10 s. The displayed approximate 19 s crossing is an interpolation-based diagnostic; the operational two-window confirmation rule must still be applied before recording an alarm lead time. Research on temporal warning evaluation distinguishes score-threshold crossing from an actionable confirmed alarm [31]. Figure 6(b) assigns false-negative shares of 41%, 17%, 21%, 13%, and 8% to loading bays, passing loops, switch areas, blind curves, and the main drift. False-positive shares are 22%, 34%, 19%, 11%, and 14%, respectively. These are compositions of the two error pools, not per-scenario error rates. Loading bays dominate missed warnings, but their exposure may also be larger. Passing-loop false positives suggest a need to distinguish permitted waiting from conflicting reservations. Evaluating that explanation requires scenario event counts and switch-release timestamps, rather than interpreting the shares as direct measures of scenario difficulty.

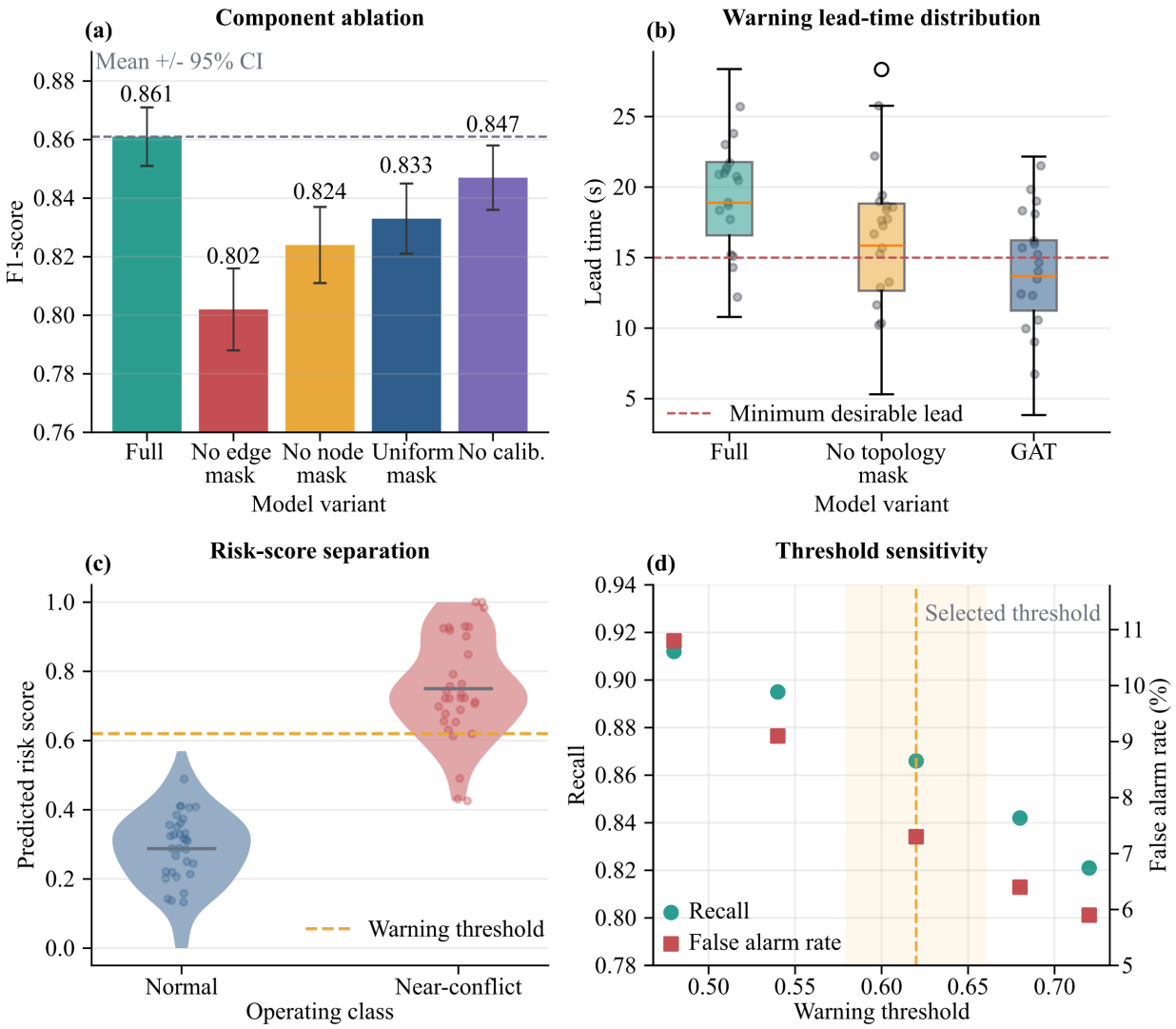


Figure 5. Ablation and threshold analysis: (a) component ablation; (b) warning lead-time distribution; (c) risk-score separation; (d) threshold sensitivity

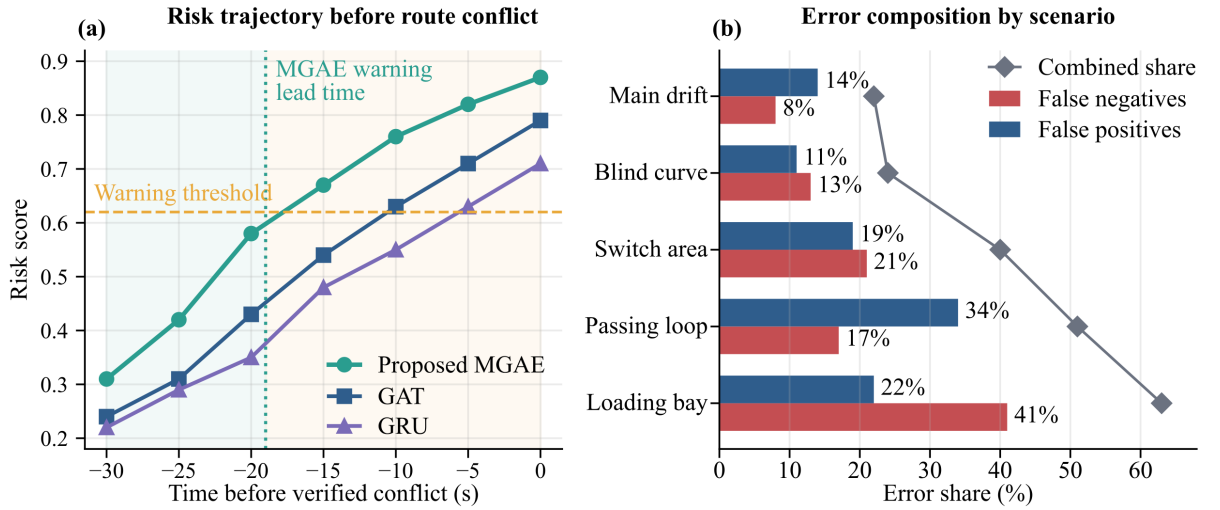


Figure 6. Scenario-level operating analysis: (a) risk-score trajectories before a route conflict; (b) error composition by scenario type

Figure 7(a) places ordinary, high-density normal, and near-conflict examples in a two-dimensional embedding projection. The high-density and near-conflict regions overlap, illustrating why a visually compact cluster cannot itself validate classification. Projection methods can distort neighborhood distances, so quantitative separation must be assessed in the original representation [32]. Figure 7(b) reports reconstruction error at 0, 40, 80, 120, and 160 epochs.

Edge-conflict error falls from 0.213 to 0.121, a reduction of approximately 43.2%, while locomotive-state, occupancy, and switch errors decrease from 0.142 to 0.066, 0.128 to 0.061, and 0.154 to 0.079. The persistent ordering suggests that route interactions are harder to reconstruct than individual states under this objective. Figure 7(c) compares label-noise levels of 0%, 2%, 5%, 10%, and 15%. At 10%, proposed-model F1 declines from 0.861 to 0.817, while graph attention declines from 0.774 to 0.691. The respective losses are 0.044 and 0.083. Random flipping is only a limited stress test: disagreement about ambiguous near-conflicts can depend on scenario and timing rather than being independent of the graph. The plotted difference therefore characterizes the stated perturbation, not every form of annotation uncertainty.

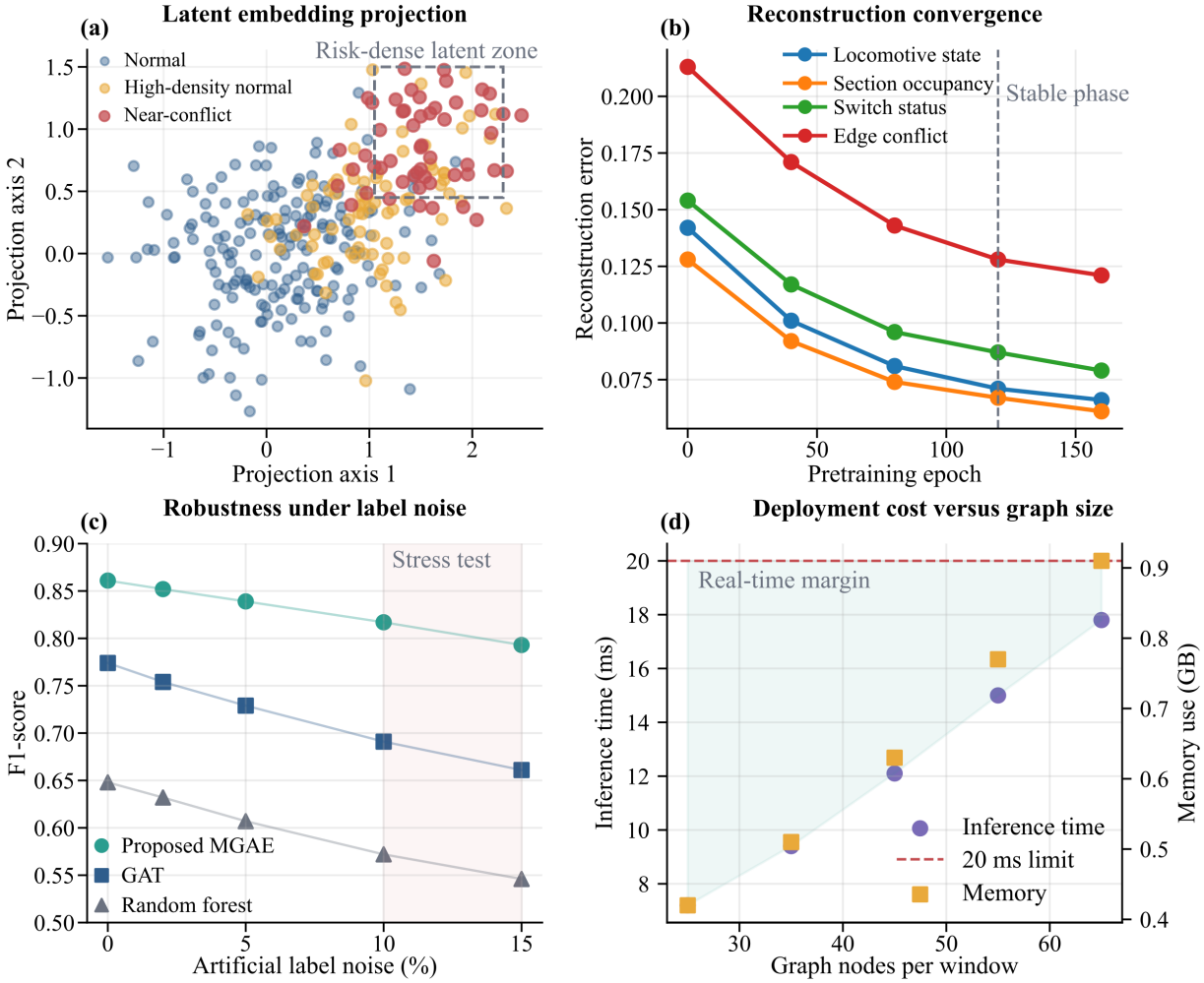


Figure 7. Representation and deployment diagnostics: (a) embedding projection; (b) reconstruction error by attribute group; (c) robustness under label noise; (d) inference time versus graph size

Figure 7(d) reports inference times of 7.2, 9.4, 12.1, 15.0, and 17.8 ms for graphs with 25, 35, 45, 55, and 65 nodes. Memory use increases over the same sizes from 0.42 to 0.51, 0.63, 0.77, and 0.91 GB. The measured inference times remain below the displayed 20 ms diagnostic limit, leaving 2.2 ms of margin at the largest tested size. Industrial monitoring research treats end-to-end latency as including acquisition, preprocessing, communication, and decision handling, rather than model execution alone [33]. Accordingly, these measurements cover model execution only and do not establish the latency of the full warning pipeline. Graph construction, communication, and two-window confirmation may add delay. The source does not report benchmark hardware, repeated timing runs, or high-percentile latency; those details should be added before making a real-time deployment claim. Memory capacity should also be verified under representative concurrent load, and route-map changes require a version check before inference.

5. Conclusion

This study evaluated a small-sample Masked Graph Autoencoder for underground locomotive collision-risk prediction using graph reconstruction and a limited labeled subset. On the reported haulage-network evaluation, the method achieved an F1-score of 0.861, AUPRC of 0.884, balanced accuracy of 0.902, a reported false-alarm rate of 7.3%, and mean warning lead time of 18.9 s at the 20% labeled condition. These are results for the stated dataset and protocol, not a deployment guarantee.

1. At the 20% labeled condition, the proposed model exceeded the graph-attention baseline by 0.087 in F1 and 0.070 in AUPRC, while reporting balanced accuracy of 0.902. The 7.3% false-alarm figure and 18.9 s mean lead time require interpretation with the metric denominator and alarm-confirmation rule stated explicitly.

2. The largest reported advantage occurred at the 5% labeled condition, where the proposed model achieved an F1-score of 0.742, compared with 0.628 for graph attention and 0.584 for the recurrent baseline. Removing edge masking reduced the full-model F1-score from 0.861 to 0.802; this ablation is evidence of sensitivity to that component in the evaluated setup, but does not by itself establish statistical significance or causality across datasets.

3. The approach is a candidate for settings with abundant unlabeled operating windows and few verified risk labels. Its evaluation is limited by one haulage network, a small number of positive windows, incomplete exposure-normalized error reporting, and protocol details that require fuller disclosure. The results support further controlled evaluation rather than a claim of general suitability or label-free prediction.

Future work should evaluate the method across multiple mines and independent conflict episodes, report repeated-run uncertainty and exposure-normalized safety metrics, and measure end-to-end alert latency under representative system load. Online graph updates and recalibration should be tested under changes in route layout, communication quality, and operating schedules.

6. Declarations

6.1 Author Contributions

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

6.2 Funding and Acknowledgments

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

6.3 Institutional Review Board Statement

Not applicable.

6.4 Informed Consent Statement

Not applicable.

6.5 Date Availability

Not applicable.

6.6 Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

6.7 Declaration of generative AI and AI-assisted technologies

During the preparation of this manuscript, the authors utilized Grammarly for linguistic polishing, logical organization and draft revision. The generative AI tool was not involved in research design, data collection, data analysis, or the equation of core conclusions. All outputs generated by AI were reviewed, revised and verified manually by the authors. The authors take full responsibility for all content of this manuscript.

7. References

[1] Cacciuttolo, C., Atencio, E., Komarizadehasl, S., & Lozano-Galant, J. A. (2024). Internet of Things Long-Range-Wide-Area-Network-Based Wireless Sensors Network for Underground Mine Monitoring: Planning an Efficient, Safe, and Sustainable Labor Environment. Sensors, 24(21), 6971. https://doi.org/10.3390/s24216971

[2] Tejada Peralta, A. F. P., Bakzadeh, R., Siahidouzazar, S., & Roghanchi, P. (2026). Automated Object Detection and Change Quantification in Underground Mines Using LiDAR Point Clouds and 360° Image Processing. Applied Sciences, 16(5), 2337. https://doi.org/10.3390/app16052337

[3] Ni, Y., Huo, J., Hou, Y., Wang, J., & Guo, P. (2024). Detection of Underground Dangerous Area Based on Improving YOLOV8.

Electronics, 13(3), 623. https://doi.org/10.3390/electronics13030623

[4] Zeng, B., Yang, X., Hu, P., Wang, Y., Dong, H., Gong, D., & Ye, X. (2024). Towards a digitally enabled intelligent coal mine integrated energy system: Evolution, conceptualization, and implementation. Sustainable Energy Technologies and Assessments, 73, 104128. https://doi.org/10.1016/j.seta.2024.104128

[5] Zvarivadza, T., Mpanza, M., Adebisi, J., Eniowo, O. D., Onifade, M., Lawal, A. I., & Khandelwal, M. (2026). Modern mining sustainability: An integrated framework for technology innovation, operational trade-offs, and implementation barriers. Green and Smart Mining Engineering, 3(3), 357-385. https://doi.org/10.1016/j.gsme.2026.03.004

[6] Chen, X., Yang, C., Hu, H., Gao, Y., Zhu, Q., & Shao, G. (2024). A Hybrid DWA-MPC Framework for Coordinated Path Planning and Collision Avoidance in Articulated Steering Vehicles. Machines, 12(12), 939. https://doi.org/10.3390/machines12120939

[7] Zhang, C., Yang, X., Zhou, R., & Guo, Z. (2024). A Path Planning Method Based on Improved A* and Fuzzy Control DWA of Underground Mine Vehicles. Applied Sciences, 14(7), 3103. https://doi.org/10.3390/app14073103

[8] Sun, Y., Zhang, L., Liu, J., Xu, Y., & Li, X. (2025). Autonomous Driving of Trackless Transport Vehicles: A Case Study in Underground Mines. Sensors, 25(10), 3189. https://doi.org/10.3390/s25103189

[9] Miao, W., & Zhao, X. (2024). Traffic Congestion Scheduling for Underground Mine Ramps Based on an Improved Genetic Scheduling Algorithm. Applied Sciences, 14(21), 9862. https://doi.org/10.3390/app14219862

[10] Abdukodirov, A., & Benndorf, J. (2025). Recent Developments in Path Planning for Unmanned Ground Vehicles in Underground Mining Environment. Mining, 5(2), 33. https://doi.org/10.3390/mining5020033

[11] Tokarczyk, J., Szewerda, K., Michalak, D., & Orzech, Ł. (2026). A Concept of an Emergency Braking Device for a Mine Suspended Monorail Travelling at an Increased Speed. Applied Sciences, 16(3), 1338. https://doi.org/10.3390/app16031338

[12] Muthineni, K., Artemenko, A., Abode, D., Vidal, J., & Nájar, M. (2025). PosGNN: A Graph Neural Network Based Multimodal Data Fusion for Indoor Positioning in Industrial Non-Line-of-Sight Scenarios. IEEE Open Journal of Vehicular Technology, 7, 15-26. https://doi.org/10.1109/ojvt.2025.3630970

[13] Yu, Z., Liu, Q., & Qian, Y. (2026). A UWB Underground Mine Positioning Algorithm Based on Graph Neural Network. Applied Sciences, 16(14), 7105. https://doi.org/10.3390/app16147105

[14] Shi, J., Wang, S., Qu, P., & Shao, J. (2024). Time series prediction model using LSTM-Transformer neural network for mine water inflow. Scientific Reports, 14(1), 18284. https://doi.org/10.1038/s41598-024-69418-z

[15] Zhang, Y., Zhao, Y., Li, Z., Cheng, X., Wang, Y., Kotevska, O., Yu, P. S., & Derr, T. (2024). A Survey on Privacy in Graph Neural Networks: Attacks, Preservation, and Applications. IEEE Transactions on Knowledge and Data Engineering, 36(12), 7497-7515. https://doi.org/10.1109/tkde.2024.3454328

[16] Fu, Z., Zhu, H., Zhao, Y., Huan, R., Zhang, Y., Chen, S., & Pan, Y. (2024). GMAEEG: A Self-Supervised Graph Masked Autoencoder for EEG Representation Learning. IEEE Journal of Biomedical and Health Informatics, 28(11), 6486-6497. https://doi.org/10.1109/jbhi.2024.3443651

[17] Zhu, X., Zhang, Q., Cui, H., Li, C., & Zhao, Z. (2026). M3GAE: Motif-guided multi-view masked graph autoencoder for graph representation learning. Pattern Recognition, 180, 114495. https://doi.org/10.1016/j.patcog.2026.114495

[18] Pan, K., & Lai, D. (2026). Empower Multiscale Masked Graph Autoencoder With Graph Structural Entropy. IEEE Transactions on Computational Social Systems, 13(3), 3012-3027. https://doi.org/10.1109/tcss.2026.3658674

[19] Zhang, G., Zuo, E., Yan, Z., Chen, C., Chen, C., & Lv, X. (2025). Rethinking link prediction: A multi-scale graph masked autoencoder. Neurocomputing, 664, 132115. https://doi.org/10.1016/j.neucom.2025.132115

[20] Tu, B., He, B., He, Y., Zhou, T., Liu, B., Li, J., & Plaza, A. (2025). Self-Supervised Masked Graph Autoencoder for Hyperspectral Anomaly Detection. IEEE Transactions on Image Processing, 34, 6714-6729. https://doi.org/10.1109/tip.2025.3620091

[21] Li, Y., Zhang, X., Guan, S., Ma, G., & Kong, Y. (2025). Topology-Guided Graph Masked Autoencoder Learning for Population-Based Neurodevelopmental Disorder Diagnosis. IEEE Transactions on Neural Systems and Rehabilitation Engineering, 33, 1550-1561. https://doi.org/10.1109/tnsre.2025.3562662

[22] Liu, Q., Ye, J., Liang, H., Sun, L., & Du, B. (2024). TS-MAE: A masked autoencoder for time series representation learning. Information Sciences, 690, 121576. https://doi.org/10.1016/j.ins.2024.121576

[23] Jia, M., Xu, D., Yang, T., Yao, Y., & Liu, Y. (2024). Graph-guided masked autoencoder for process anomaly detection. Process Safety and Environmental Protection, 186, 1345-1357. https://doi.org/10.1016/j.psep.2024.04.052

[24] Zhang, S., Xi, P., Jiang, M., Zhang, G., & Cheng, D. (2025). Latent Representation Learning for Attributed Graph Anomaly Detection. ACM Transactions on Knowledge Discovery from Data, 19(7), 1-22. https://doi.org/10.1145/3733604

[25] Hu, Y., Chen, C., Deng, B., Lai, Y., Lin, H., Zheng, Z., & Bian, J. (2024). Decoupling Anomaly Discrimination and Representation Learning: Self-supervised Learning for Anomaly Detection on Attributed Graph. Data Science and Engineering, 9(3), 264-277. https://doi.org/10.1007/s41019-024-00249-8

[26] Sattarzadeh, A. R., Pathirana, P. N., & Palaniswami, M. (2024). Implicit sensing self-supervised learning based on graph multi-pretext tasks for traffic flow prediction. Neural Computing and Applications, 37(2), 1041-1066. https://doi.org/10.1007/s00521-024-10461-2

[27] Fu, J., Zhou, W., & Chen, Z. (2024). Bayesian graph convolutional network for traffic prediction. Neurocomputing, 582, 127507. https://doi.org/10.1016/j.neucom.2024.127507

[28] Gao, X., Jiang, X., Haworth, J., Zhuang, D., Wang, S., Chen, H., & Law, S. (2024). Uncertainty-aware probabilistic graph neural networks for road-level traffic crash prediction. Accident Analysis & Prevention, 208, 107801. https://doi.org/10.1016/j.aap.2024.107801

[29] Du, Q., Wang, X., Yin, S., Li, L., & Ning, H. (2024). Social Force Embedded Mixed Graph Convolutional Network for Multi-Class Trajectory Prediction. IEEE Transactions on Intelligent Vehicles, 9(9), 5571-5580. https://doi.org/10.1109/tiv.2024.3352180

[30] Xia, T., & Chen, H. (2024). A Survey of Autonomous Vehicle Behaviors: Trajectory Planning Algorithms, Sensed Collision Risks, and User Expectations. Sensors, 24(15), 4808. https://doi.org/10.3390/s24154808

[31] Liu, Z., Zou, G., Wang, T., Tu, M., Wang, H., & Li, Y. (2026). Learning and predicting traffic conflicts in mixed traffic: A spatiotemporal graph neural network with manifold similarity learning. Expert Systems with Applications, 309, 131183. https://doi.org/10.1016/j.eswa.2026.131183

[32] López-López, E. M., Bru-Cordero, O. E., & Correa-Álvarez, C. D. (2026). Rare-Event Road-Traffic Fatality Prediction: A Reproducible Machine-Learning Benchmark with Time-Aware Validation, Calibration, and Interpretability. Future Transportation, 6(4), 146. https://doi.org/10.3390/futuretransp6040146

[33] Shi, J., Li, J., Tam, W. C., Gardoni, P., & Usmani, A. S. (2024). Physics_GNN: Towards Physics-informed graph neural network for the real-time simulation of obstructed gas explosion. Reliability Engineering & System Safety, 256, 110777. https://doi.org/10.1016/j.ress.2024.110777