Machine Learning for Robust Network Design: A New Perspective

Chenyi Liu, Vaneet Aggarwal, Tian Lan, Nan Geng, Yuan Yang, and Mingwei Xu

The authors give a selective overview of current work on robust network design and show that failure evaluation provides a common kernel to improve the tractability and scalability of existing solutions.

Abstract

With the rapid growth of backbone networks and data center networks, ensuring network robustness under various failure scenarios has become a key challenge in network design. The combinatorial nature of failure scenarios in data plane, control plane, and management plane seriously challenges existing practice on robust network design, which often requires verifying the designed network's performance by enumerating all possible failure combinations. Meanwhile, machine learning (ML) has been applied to many networking problems and has shown tremendous success. In this article, we show a general approach to leveraging machine learning to support robust network design. First, we give a selective overview of current work on robust network design and show that failure evaluation provides a common kernel to improve the tractability and scalability of existing solutions. Then we propose a function approximation of the common kernel based on graph attention network (GAT) to efficiently evaluate the impact of various potential failure scenarios and identify critical failures that may have significant consequences. The function approximation allows us to obtain new models of three important robust network design problems and to solve them efficiently by evaluating the solutions against a pruned set of critical failures. We evaluate our approach in the three use cases and demonstrate significant reduction in time-to-solution with minimum performance gap. Finally, we discuss how the proposed framework can be applied to many other robust network design problems.

Introduction

As networks grow in scale and complexity, failures are becoming a common and frequent event in both wide area networks (WANs) and data center networks (DCNs) [1]. The combination of various failures (including hardware faults, software bugs in the control plane, and misconfiguration by the network operator) may cause unpredictable paralysis of the entire network [2]. Recently, increasing research efforts have considered tackling the problem from different aspects. Some studies consider the network planning stage [3, 4] that minimizes the upgrade cost for optical and IP layer while ensuring network availability under possible failure scenarios. Some studies focus on the traffic engineering perspective [5–8] that makes fault-tolerant traffic management decisions against diverse failure combinations. There are also research efforts on robust network validation [9] that quantifies worst-case network performance given a set of possible failures. We jointly name these problems, including but not limited to robust network planning, robust traffic engineering, and robust network validation, as *robust network design* problems. In order to support robust network design, existing approaches often rely on model-based mixed-integer optimization, which is hard to scale to large-scale modern networks.

In robust network design, ensuring network availability under diverse failure scenarios is crucial but computationally challenging due to the exponential growth of failure scenarios with network size [6, 9, 10]. Model-based solutions often use approximation and relaxation techniques to reduce the failure space, but these may not be suitable for all problems and can lead to performance issues. In this article, we show that machine learning can offer a new approach to address the core challenge in robust network design by identifying critical failures that significantly impact network performance from a large set of possible failure scenarios. By accurately and efficiently evaluating failure impacts, machine learning provides a common kernel that can benefit many robust network design problems.

Graph neural networks (GNN) are deep learning models for processing graph data and have shown success in extracting network features [3] and modeling network performance [11]. This article proposes using graph attention networks (GAT) [12], a state-of-the-art GNN model, for efficient and accurate failure impact evaluation. The GAT-based approach allows for one-shot inference, avoiding time-consuming network behavior simulations. With this approach, many computationally prohibitive robust network design problems can be recast and accelerated.

The main contributions of this article are highlighted as follows:
- Failure evaluation is identified as a common kernel to enhance current large-scale robust network design problems based on the selective overview of existing approaches.
- A GAT-based failure evaluation function that takes network topology, traffic demand, routing decisions, and target failure scenarios as

C. Liu, N. Geng, Y. Yang, and M. Xu were supported in part by the National Key R&D Program of China (2022YFB2901303), the National Natural Science Foundation of China under Grant (62132004, 62221003 and 61832013). V. Aggarwal and T. Lan were supported in part by Cisco, Inc.

Digital Object Identifier: 10.1109/MCOM.002.2200670

Chenyi Liu and Nan Geng are with Tsinghua University, China; Yuan Yang and Mingwei Xu corresponding authors) are with Tsinghua University, China and also with with Zhongguancun Laboratory, Beijing, China and Beijing National Research Center for Information Science and Technology, Beijing, China; Vaneet Aggarwal is with Purdue University; Tian Lan is with George Washington University.

inputs and generates failure impact predictions in a one-shot calculation is proposed in a high-scalable fashion.
- With the GAT-based failure evaluation function, several robust network design problems, including robust network validation, network upgrade optimization, and fault-tolerant traffic engineering, are recast and significantly accelerated.
- Broader applications of ML-based failure evaluation in robust network design are discussed.

We organize the remainder of this paper as follows. The next section presents a selective overview of existing work on robust network design and identifies the common kernel. Then, we introduce our proposed approach that leverages GATs to develop and implement the common kernel of failure evaluations, enabling us to recast three important robust network design problems. Finally, we present evaluation results and discuss future directions.

Robust Network Design

A Selective Overview

Designing a robust network is challenging due to hardware failures, configuration, and software bugs, which are inevitable even with good engineering practices [2]. Multiple failures in the network can cascade and cause severe and unpredictable effects, making it difficult to protect the network from all possible failure combinations. Link failures and router/switch failures are common considerations in existing work, while data consistency and control plane failures are also considered in recent research. We present the representative research efforts of robust network design as follows.

Robust network validation is crucial for identifying potential weaknesses and guiding future network upgrades. However, validating all possible failure scenarios is infeasible for large-scale networks with numerous combinations of failures. Existing research has attempted to solve this problem, such as using a two-stage max-min optimization problem to find worst-case performance [9]. However, scalability and optimality remain challenges for large-scale networks. Another approach is to prune the failure space using routing strategies and failure probability models [13], but a scalable algorithm that can be applied to various scenarios is still lacking.

Robust traffic engineering aims to find the best traffic management strategy while ensuring availability. Most existing approaches enumerate potential failure scenarios and consider each one separately, resulting in high time costs for large-scale networks [7]. Some approaches relax integer variables into continuous ones and use approximation methods [10, 14], but this incurs significant computation overhead or performance degradation [5, 6]. Additionally, existing approaches have only been successful in optimizing specific problems rather than providing a general solution.

Robust network planning determines short-term and long-term future network designs, including optical and IP layers, according to traffic forecasts [4]. Large-scale WAN planning problems are usually solved with ILP minimizing network

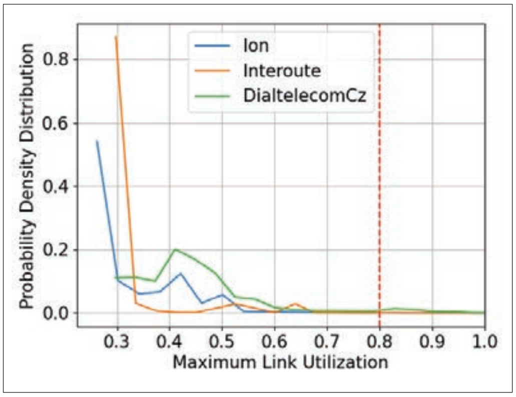


FIGURE 1. Distribution of MLU increase on large-scale network topology under 2 simultaneous link failures. MLU under failure scenarios are normalized by the MLU under the worst-case failure scenario.

cost while ensuring availability under a set of failure combinations [3]. However, robust network planning considers each failure scenario separately in the optimization model, limiting the number of failure scenarios that can be protected due to the quickly-growing complexity of their combinations within the ILP problem.

Challenges and A Common Kernel

Recent works [2, 3] by Microsoft and Facebook show that modern wide-area-network (WAN) are becoming larger and larger with thousands of routing nodes and are capable of taking larger-scale traffic flow. Moreover, the topology size keeps growing at a rate of 20% per year with no end in sight [3]. Large and complex network topology makes it difficult to design a robust network under millions of failure scenarios. In this section, we firstly show that the huge failure space is the key challenge to robust network design. Then, we show that evaluating failure impacts efficiently and focusing on the critical failures with great impact on the network could be a common kernel to overcome the key challenge of robust network design problems.

Existing works for robust network design usually build an optimization model to optimize the network performance under various failure scenarios. In particular, robust network validation solves multi-commodity flow (MCF) problems for all the possible failure scenarios, while robust network planning and robust traffic engineering solve LP or ILP problems with congestion-free constraints for each failure scenario. However, the number of failure combinations grows exponentially with the size of the network, which makes many robust network design problems unsolvable. For instance, many recent proposals studied ensuring the network performance under f simultaneous link failures. For a network topology with m links, only considering f simultaneous link failure combinations would bring $O(m^f)$failure scenarios under consideration, causing a super-linear growth in the LP/ILP problems with the network scale. Moreover, the solution time to such LP/ILP problems also increases super-linearly with the number of decision variables and constraints [4]. The two factors mentioned above jointly make robust network design problems hard to solve with increasingly large and complex network topologies and failure scenarios in the real

Existing works for robust network design usually build an optimization model to optimize the network performance under various failure scenarios.

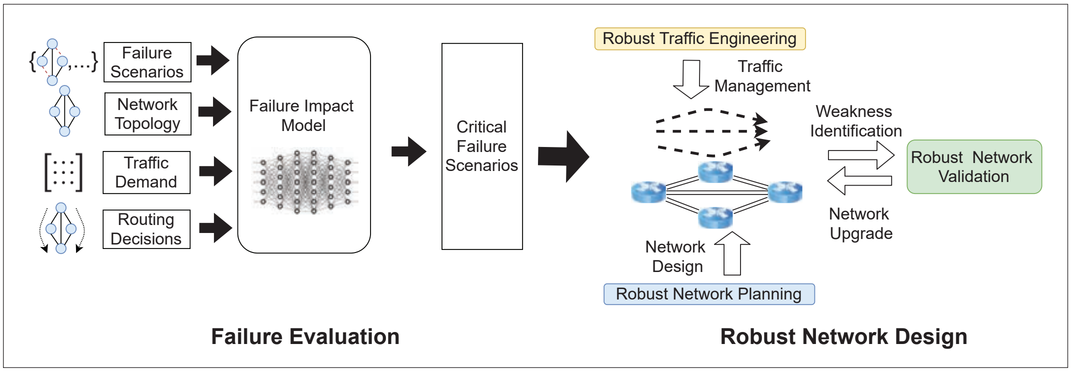


FIGURE 2. General perspective of our proposed machine learning-based approach for robust network design.

world. In particular, even for a 100-node network, the time to solve a fault-tolerant traffic engineering problem via linear programming (LP) could take several weeks, making it unacceptable in any dynamic environment where network topology and traffic characteristics are time-varying. Furthermore, the combination of failures could be much more complicated in practice. The complicated failures make it even harder to enumerate all the failures and guarantee the availability under failures in a typical LP/ILP problem.

Although the number of possible failure combinations is huge, we could prune the failure space based on some key indicators. For instance, some existing research [13] only considers the failure combinations with high occurring probability. Unfortunately, obtaining the exact probabilistic models of network components in practice is often difficult. In this article, we show that failure impact could be an effective indicator to figure out the critical failure scenarios and prune the considering failure space.

Most existing robust network design problems validate/optimize the worst-case network performance (e.g., maximum link utilization (MLU)) under failure scenarios. In other words, they focus on improving the network availability under the failure scenarios that greatly impact the network. In this article, we use the increase of MLU under a failure scenario to measure the failure impact. MLU is a popular performance metric in existing works for robust network design to measure the network congestion level [9]. Thus MLU increase indicates the degree of congestion increase in the network under failure scenarios, and is independent of the input traffic load level since it is normalized by MLU in the non-failure scenario. We note that such a metric for failure impact is representative of many robust network design problems, and can be extended to a unified measure of failure impact like latency, throughput, etc. We analyze the failure impact and find that only a small subset of failure scenarios greatly impact the large-scale network. Fig.ure 1 shows the distribution of failure impact (i.e., MLU increase) for three large-scale real-world network topologies with more than 100 nodes. It turns out that only 0.19%, 0.03%, and 3.43% failure scenarios on Ion, Interoute, and DialtelecomCz, respectively, cause significant impact (i.e., more than 80% of worst-case failure impact) on the network availability. It implies that by providing an approximation of the failure impact evaluation function, we could prune many unimportant failure scenarios and focus only on a small subset of critical failure scenarios with great failure impacts in robust network design.

Unfortunately, modeling the failure impact is quite a challenging task. For instance, in a theoretically optimal setting we need to solve an MCF problem for each failure scenario to simulate the failure impact. Moreover, practical network failures may be caused by a combination of failures in the data plane, control plane, and management plane, which makes it even more difficult to model the impact of a failure combination. The complexity and the huge number of possible failure combinations make it quite difficult to figure out the critical failure scenarios that cause significant impacts on the network. In this paper, we explore the potential of machine learning in resolving the common core of evaluating the failure impact and detecting critical failure scenarios for robust network design.

Enhancing Robust Network Design with Machine Learning-Based Failure Evaluation

General Perspective

In this article, we show a general approach to resolve robust network design problems using machine learning-based failure evaluation. We show the general perspective of our approach in Fig. 2. In the general perspective, we resolve a robust network design problem in two steps. First, we design a machine learning-based function to predict the impact of target failure scenarios and figure out critical failure scenarios with significant impacts. In general, the failure impact under a given failure scenario is determined by network topology, traffic demand, and routing decision. The machine learning-based function takes the target failure scenarios, network topology, traffic demand, and routing decision as input, and outputs predicted impact of the target failure scenarios. With the failure evaluation results, we can select a small subset of critical failure scenarios from the full failure set. Such a failure evaluation algorithm should have the characteristics described in the following:

- **High computational efficiency:** The algorithm should have low time cost and memory use, and keep a low overhead increase when the topology scale increases.
- **High accuracy:** The algorithm should accurately predict the impact of target failure scenarios, especially for potentially critical failure cases.

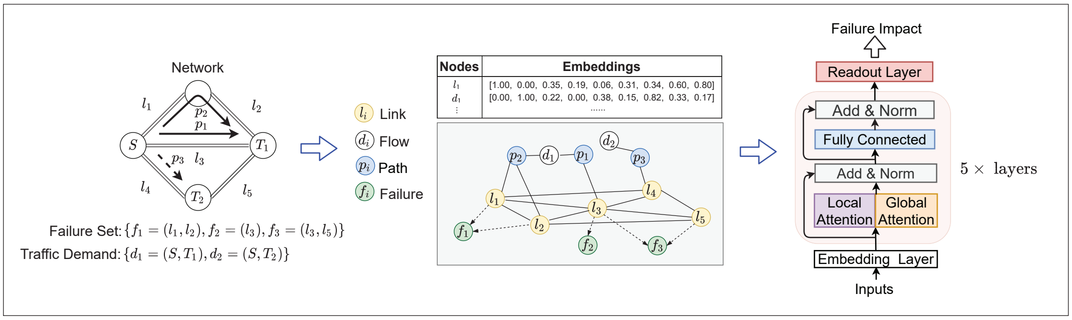


FIGURE 3. Overall structure of GAT-based failure impact prediction model.

- **Good generalization:** The algorithm should have good generalization to unseen network topologies, traffic demands, different failure types, and other possible application scenarios.

Second, we recast three typical robust network design problems using the failure impact function and then enhance the tractability of the recast problems with the selected critical failure scenarios.

To simplify the problem, we aim to evaluate the impact of multiple link failure scenarios and design networks robust to such link failures in this article. However, we note that such a general framework can be extended to solve many other robust network design problems with diverse types of failures.

A Deep Learning Mechanism for Link Failure Evaluation

In this section, we propose a generalizable failure evaluation approach [15] based on a graph attention network (GAT) for multiple link failures. We intuitively show the major insights to design a generic failure impact model and then illustrate the design of the proposed GAT-based model.

Major Insights to Failure Impact: With the analysis to the rerouting process for different routing schemes (e.g., optimal MCF and OSPF), we show some key insights to failure impact in the following:

- **Critical indicators:** The link utilization and traffic volume passing through the link under non-failure routing decision are two important features indicating a critical link failure. In particular, a link with higher link utilization and traversed traffic volume may cause a greater impact on the network.
- **Locality:** Network topology and traffic demand in current real world WAN are clustered locally. When a failure occurs, finding a reroute locally to bypass the failed link could save the overall network capacity and cause a smaller impact on the network. Thus, a local understanding of the topology structure near a failed link is important to predict the impact of the failed link.
- **Long flow reroute:** In situations where large and long flows encounter a failed bottleneck link, rerouting them locally may not be enough. Instead, a global understanding of the network and traffic routes is needed to assess the impact of rerouting the flow through longer alternative paths.

Network failures can cause congestion in both neighboring and distant parts of the network due to traffic rerouting. To accurately estimate the impact of these failures, it is important to consider global network information such as topology and capacity variations. To address this issue, we have designed a GAT-based model that uses critical indicators as input and employs local and global attention mechanisms to provide a comprehensive understanding of the network state.}

Graph Attention Networks: The Graph Attention Network (GAT) \cite{brody2021attentive} is a type of graph neural network that leverages attention mechanisms to process graph data. Specifically, GATs embed input states into a directional graph, where each node in the graph is associated with a feature vector that represents information such as link capacity, link utilization, flow traffic demand, and node type. The key idea behind GATs is neighborhood aggregation, which involves computing a linear combination of the feature vectors of a central node and its neighbors for each node in the graph. The GAT accomplishes this by using an attention mechanism to learn weights that determine the importance of each neighbor's features for the central node.

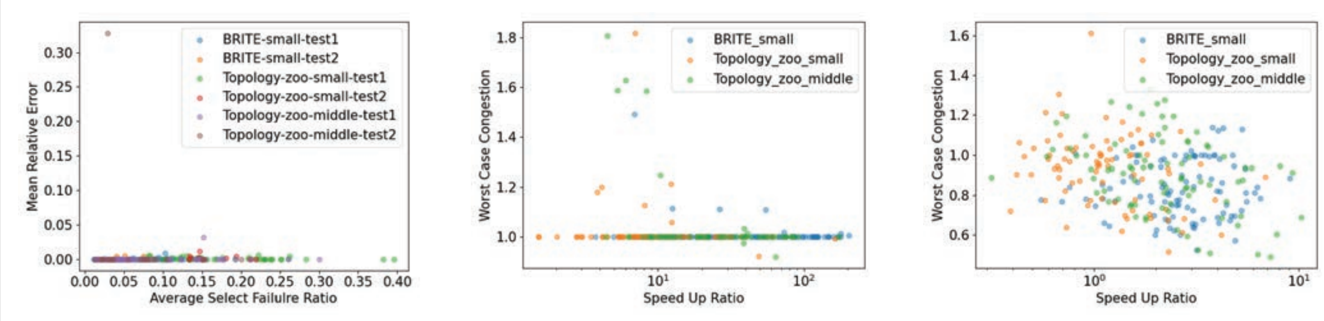
The attention mechanism in GATs is based on computing a set of attention coefficients e_{ij} that indicate the importance of node *i*'s features for the central node *j*'s. These coefficients are obtained by taking the dot product of a shared parameter vector, which serves as trainable parameter in the GAT model, and a concatenation of node *i*'s features and node *j*'s features. This produces a single scalar value, which is then passed through a softmax function to ensure that the coefficients sum to 1 across all of node *j*'s neighbors.

Once the attention coefficients have been computed, the GAT uses them to compute a weighted sum of the feature vectors of the central node and its neighbors. This weighted sum is then passed through a non-linear activation function to obtain the final output for the central node.

GATs' ability to capture the influence of each node's neighbors in the graph makes it suitable for embedding and representing network features in failure evaluation. It enables identification of key information of local network structure and traffic assignments (represented as graph data), e.g., how neighboring links are affected by a failed link in the network.

GATs offer the advantage of handling graphs of varying sizes and structures, adapting to different topologies by learning to focus on relevant parts through its attention mechanism. As a result, GATs are powerful tools for learning graph-structured data representations and have the potential for various applications, including robust network design and failure impact evaluation.

Failure Evaluation Model Design: In this article, we embed the correlation between traffic flows, routing paths, links and target failures in the



(a) Robust network validation (b) Network upgrade optimization (c) Fault-tolerant traffic engineering

FIGURE 4. Performance of ML-enhanced approach on three robust network problems. Each dot in the figure represents a topology in the testing dataset. The worst-case congestion (i.g., MLU) is normalized by the results of the baseline optimization approach, and the speed-up ratio is obtained by dividing the time overhead of baseline optimization problem by the time overhead of our proposed ML-enhanced algorithm: a) robust network validation; b) network upgrade optimization; c) fault-tolerant traffic engineering.

network into a graph. Then, we leverage GAT and its attention mechanism to weigh the influence from neighboring nodes in the input graph and to represent the key information for inferring the failure impact in each failure scenario. We also introduce an attention mechanism to weigh the failure influence from a broader aspect. In addition to aggregating information from neighbors in standard GAT, a global attention mechanism enables each link to further aggregates information and weigh the failure influence from all the other links in the network. Thus, the resulting model could better evaluate the impact of failures by combining global and local information.

We first exhibit the transformation process to convert the input network topology, traffic demand, routing decisions, and target failure scenarios as a graph-based input state. Then we use GATs to extract key features of network state (e.g., network structure and flow assignment) for each failure scenario and infer the failure impact based on the extracted features.

Model Input: To construct a suitable input graph for our GAT-based algorithm, we begin by transforming the original network topology, as shown in Fig. 3. First, each link in the original topology is transformed into a node, and a link is added between two nodes in the transformed graph if their corresponding links share a common endpoint in the original topology. Second, we model traffic demand and the routing decision of each flow under a non-link-failure scenario on the transformed graph. For each flow, we build up edges between the flow node and the corresponding path nodes, representing the routing paths of a flow. Further, for each routing path, we build up edges between the path node and the corresponding link nodes. Finally, we model the target failure scenarios. For each failure scenario, we build up the edges from the nodes of corresponding failed links to the failure node. The failure nodes aggregate link states for the final failure impact prediction. For the four types of nodes in the input graph, we design the initial state craftily to embed the link attributes, input traffic demand, and routing decision into the input graph.

Model Design: GATs provide a natural approach for estimating failure impacts through function approximation. Our proposed estimator, as illustrated in Fig. 3, comprises an input embedding layer, five local-global attention layers, and a readout layer. }Initially, the input states undergo embedding via a two-layer fully-connected deep neural network. The embedded input state then passes through five local-global layers consecutively, where feature extraction and failure impact modeling take place. Each local-global attention layer employs a graph attention mechanism to obtain a local understanding of the graph-based input. This one-hop attention process captures the local graph structure information and facilitates the link node in estimating the failure impact using the local understanding. Moreover, we have a global attention mechanism that estimate the link failure influence and aggregate information among all the link nodes with no neighbor constraints, enabling the link node to benefit from the global understanding of the input network state. The local and global attention layers' output is combined using a fully-connected layer. We incorporate residual and normalization mechanisms in each local-global attention layer to support a stack of more layers. Finally, the attention layers' aggregated representations of the failure combination nodes serve as the input for the readout layer to predict the failure impact.

Training and Inference: We train a general model over a large dataset containing a number of different network topologies. With the trained GAT-based model, we could evaluate the impact of all the target failure scenarios in a one-shot inference and choose the critical failure scenarios efficiently.

Application Scenarios

In this section, we show how to use ML-based failure evaluation to recast and solve three important robust network design problems, namely network robust validation [9], network upgrade optimization [4, 9], and fault-tolerant traffic engineering [14].

Robust network validation: Robust network validation needs to find the worst-case failure scenarios that have the most severe impact on certain network performance objectives. We can directly use the proposed failure impact function to identify such critical failures for scalable network validation.

Network upgrade optimization: Network upgrade optimization aims to minimize the cost of necessary link capacity upgrades subject to network congestion constraints under link failures. In fact, only failure scenarios that cause network congestion need to be considered in the network upgrade optimization problems. With such a simple pruning of non-essential failure scenarios, we can significantly reduce the size of the original optimization problem with little impact on the solution results.

Fault-tolerant traffic engineering: Besides pruning the candidate failure scenarios of the original problem, we can also design a new fault-tolerant traffic engineering algorithm using the prediction results of our proposed GAT-based failure evaluation function. In particular, we focus on protecting a small subset of critical failure scenarios while considering a vanilla load balance factor that can be modeled efficiently without concerning the vast number of failure combinations. Instead of enumerating all failure scenarios, it features a nearly-optimal rerouting strategy over a small set of critical failure scenarios while optimizing basic load balancing objectives.

We hasten to emphasize that all the three use cases are based on the GAT-based function for scalable failure evaluation. It allows robust network design problems to be formulated with respect to only a small subset of failure scenarios that have significant impacts on robustness.

Evaluation

In this section, we evaluate the performance of our proposed ML-enhanced approach in the three use cases mentioned before. We train and evaluate our model with both 100 randomly generated small topologies and real-world topologies from topology zoo. In order to test the generalization of our proposed ML-enhanced approach on unseen topologies, we split the topologies above into two parts. The major part are placed in *train* and *test1* dataset while the others are in *test2* dataset. We randomly generate several demand matrices for each topology using the gravity model. We combine a topology, a traffic matrix, and a set of link capacities as a piece of training data. In order to simulate the heterogeneous link capacities in real world, the link capacity of each link is randomly selected from 1, 2, 3, and 4 units for each data piece. We solve the MCF optimization problem to obtain the failure impact under each single and double simultaneous link failure of each data piece. We implement three optimization models with gurobi, an off-the-shelf optimizer, as baselines. For robust network validation, refer to [9]; for network upgrade optimization, refer to [9]; and for fault-tolerant traffic engineering, refer to [14].

We train a general failure impact evaluation model with *train* dataset for ten days on a server with an RTX2080 GPU, and evaluate the ML-enhanced approach in three robust network design problems over *test1* and *test2* dataset. The results are shown in Fig. 4. In robust network validation problem, the ML-enhanced algorithm estimates the worst-case network performance accurately by verifying the selected critical failure scenarios according to ML-based failure evaluation results. In network upgrade optimization problem, the ML-enhanced algorithm provides more than 10x time reduction over most topologies, and up to 200x time reduction in some medium-sized topologies while obtain the optimal solution on most test cases. In fault-tolerant traffic engineering problem, the ML-enhanced algorithm achieves up to 9x speed up for some medium-sized topologies while providing comparable performance to the optimization solution. Besides the results above, we could also improve the model performance by a domain specific (e.g., a specific topology) fast second-phase training based on the pre-trained general model, which will be discussed in our future work. Further, we note that the ML-enhanced algorithm requires much less memory and could calculate the solution for the topologies with up to 80 (107) links for network upgrade optimization (fault-tolerant traffic engineering), while the optimization problem will exceed the 256 GB sever memory limit for the topologies with more than 45 (70) links.

Outlook

In this article, we have shown that a GAT-based function approximation could accurately predict the failure impact, detect the critical failure scenarios, and enhance the scalability of three important classes of robust network design problems. We further note that such an approach using deep learning to evaluate failure impact could benefit many other robust network design problems in real world. We discuss some future directions following our proposed approach.

Incorporating multiple failure types is crucial for robust network design. Failures can happen in various network parts and layers, from router line cards to control plane software. However, modeling and understanding the impact of different failure types together can be challenging. Nevertheless, a learning-based solution can still be applied. By using function approximation and deep learning, we can efficiently predict failure impact across multiple types through one-shot inference and identify critical failure scenarios.

Considering probabilistic failures is important for robust network design. In addition to the direct impact on network performance, the probability of different failure scenarios plays a significant role [7]. To incorporate failure probability, we can filter out low-probability scenarios from our predicted critical failures. However, obtaining the probabilistic model, especially in cases with joint failure probabilities, can be challenging. Embedding a probabilistic failure model into the machine learning-based framework could become an alternative choice that requires further exploration.

More application scenarios can benefit from our learning-based approach. Although this paper focuses on three representative use cases, robust network design has a wide range of potential applications. For example, our approach could be applied to different routing strategies, such as tunnel-based TE [7], or various network performance metrics like total throughput or 99% high latency. New common cores can be identified and implemented with the support of deep learning methods. Our vision is to develop a unified framework that leverages a spectrum of function approximations as shared common cores for supporting many network design and optimization problems that need scalable solutions.

Conclusion

This work provides a nnew perspective that applies machine learning to resolve a common kernel, i.e., failure evaluation, to enhance robust network design problems. To resolve the common kernel, we propose a GAT-based algorithm to evaluate the failure impact and figure out the critical failure scenarios among multiple link failures, and apply our proposed GAT-based algorithm to solve three typical recast robust network design problems. We apply our approach to three common robust network design problems and test it on over 100 real-world network topologies, demonstrating its efficiency and versatility. Our approach has potential for future applications with different types of failures and scenarios.

References

[1] R. Govindan *et al.*, "Evolve or Die: High-Availability Design Principles Drawn from Googles Network Infrastructure," *ACM SIGCOMM*, 2016, pp. 58–72.

[2] U. Krishnaswamy *et al.*, "Decentralized Cloud Wide-Area Network Traffic Engineering with BLASTSHIELD," *USENIX NSDI*, 2022, pp. 325–38.

[3] H. Zhu *et al.*, "Network Planning with Deep Reinforcement Learning," *ACM SIGCOMM*, 2021, pp. 258–71.

[4] S. S. Ahuja *et al.*, "Capacity-Efficient and Uncertainty-Resilient Backbone Network Planning with Hose," *ACM SIGCOMM*, 2021, pp. 547–59.

[5] C. Jiang, S. Rao, and M. Tawarmalani, "PCF: Provably Resilient Flexible Routing," *ACM SIGCOMM*, 2020, pp. 139–53.

[6] Y. Chang *et al.*, "Lancet: Better Network Resilience by Designing for Pruned Failure Sets," *ACM SIGMETRICS*, vol. 3, no. 3, 2019, pp. 1–26.

[7] J. Bogle *et al.*, "TEAVAR: Striking the Right Utilization-Availability Balance in WAN Traffic Engineering," *ACM SIGCOMM*, 2019, pp. 29–43.

[8] J. Zheng *et al.*, "Sentinel: Failure Recovery in Centralized Traffic Engineering," *IEEE/ACM Trans. Net.*, vol. 27, no. 5, 2019, pp. 1859–72.

[9] Y. Chang, S. Rao, and M. Tawarmalani, "Robust Validation of Network Designs under Uncertain Demands and Failures," *USENIX NSDI,* 2017, pp. 347–62.

[10] H. H. Liu *et al.*, "Traffic Engineering with Forward Fault Correction," *ACM SIGCOMM*, 2014, pp. 527–38.

[11] M. Ferriol-Galmés *et al.*, "RouteNet-Erlang: A Graph Neural Network for Network Performance Evaluation," *IEEE INFOCOM*, 2022, pp. 2018–27.

[12] S. Brody, U. Alon, and E. Yahav, "How Attentive are Graph Attention Networks?" arXiv, 2021.

[13] S. Steffen *et al.*, "Probabilistic Verification of Network Configurations," *ACM SIGCOMM*, 2020, pp. 750–64.

[14] Y. Wang *et al.*, "R3: Resilient Routing Reconfiguration," *ACM SIGCOMM*, 2010, pp. 291–302.

[15] C. Liu *et al.*, "FERN: Leveraging Graph Attention Networks for Failure Evaluation and Robust Network Design," arXiv, 2023.

Biographies

Chenyi Liu (liucheny19@mails.tsinghua.edu.cn) received his B.Sc. degree in computer science and technology from Tsinghua University in 2019. He is currently a Ph.D. candidate at the Department of Computer Science and Technology of Tsinghua University.

Vaneet Aggarwal (vaneet@purdue.edu) received his Ph.D. from Princeton University in 2010. He is currently a Full Professor at Purdue University.

Tian Lan (tlan@gwu.edu) received his Ph.D. from Princeton University in 2010. He is currently a Full Professor at George Washington University.

Nan Geng (nan_geng@sina.com) received the Ph.D. degree from Tsinghua University in 2021.

Yuan Yang (yangyuan_thu@mail.tsinghua.edu.cn) received the B.Sc., M.Sc., and Ph.D. degrees from Tsinghua University. He is currently an Assistant Researcher at Tsinghua University.

Mingwei Xu (xumw@tsinghua.edu.cn) received the B.Sc. and Ph.D. degrees from Tsinghua University. He is currently a Full Professor at Tsinghua University.