Response to Editor and Reviewers

We would like to express our sincere thanks to the editor and reviewers for their careful reading of our manuscript and their constructive comments. We have carefully considered all suggestions and have revised our manuscript accordingly. In the revised manuscript (with changes marked), we have indicated line numbers and highlighted all modifications in pink to make them easy to identify. The content in the blue boxes in the "revision_note.docx" is the supplement we have made to the paper content according to the comments. We believe these changes have significantly improved the quality of our paper, and we are grateful for the opportunity to address these issues.

Below, we provide point-by-point responses to each reviewer's comments.

Reviewer #1

Comment 1:

Article - in some cases, a white-space between a word and a link on the references (e.g., "complexity[1]") is missing. Please, check the whole article once again.

Response:

Thank you for your careful reading. We have added appropriate white-space between words and reference links throughout the entire manuscript.

Comment 2:

Introduction - information about the organization of the paper is missing.

Response: Thank you for this helpful suggestion. We have added the organization of this paper at the end of the introduction (lines 108-117) to guide readers through the structure of our work.

> The remainder of this paper is organized as follows. Section 1 reviews related work in network tomography approaches and deep learning applications. Section 2 introduces the network model and presents our proposed framework, including the Additive Congestion Status and probing flow mechanism. Section 3 details our methodology for combining LSTM networks with adversarial autoencoders. Section 4 presents our experimental evaluations and comparative analysis. Finally, Section 5 concludes the paper and discusses future research directions.

Comment 3:

Introduction / Section 2 - the state-of-the-art (SOTA) is elaborated on general level. I have several recommendations here. Firstly, the main contributions of the paper (in detail) should be presented after a deep evaluation of SOTA. Secondly, the authors should work with more papers from the period 2020-2024. From this point of view, I can also recommend for the authors to check paper "Distributed Network Tomography Applied to Stochastic Delay Profile Estimation" in which a system model and a distributed algorithm to estimate a network delay profile with stochastic properties were introduced. I hope this work can be helpful in the further elaboration of SOTA. Otherwise, work with another paper. Thanks! Next, please, explain the main differences between this work and your previously published papers.

Response: Thank you for your detailed recommendations. We have addressed each point as follows:

1. **Regarding contribution presentation after SOTA evaluation:** We have added two additional paragraphs to present the challenges in current approaches and how our methods address them. This leads naturally into our work's contributions (lines 71-107).

Based on a simple observation in normal network scenaries, a congested link will cause all paths passing through it to become congested. Therefore, network Boolean tomography extracts the link status by reversing the binary path status. Consider the example network shown in Figure 1, where paths {y1, y2, y3} traverse through five links {x1, x2, x3, x4, x5}. In traditional Boolean tomography, a congested link causes all paths passing through it to become congested, leading to binary path states B = R ∨ X.

However, this approach is ill-posed, as multiple sets of congested links (solutions) often match the observed congestion on the end-to-end paths, limiting its diagnostic performance. It should be noted that once a path traverses multiple congested links, its packet loss rate and communication delay are much higher than those of a single congested link. For instance, in Figure 1(a), path y1 has a 9% packet loss rate traversing one congested link, while path y3 shows a 6% loss rate passing through two congested links. By leveraging this difference, we propose Additive Congestion Status A+ = R · X to quantify the number of congested links in each path. As illustrated in Figure 1(b), this additional information helps reduce the solution space and improves diagnostic accuracy, achieving better precision and recall compared to Boolean approaches.

[Our contribution follows.]

2. **Regarding recent literature (2020-2024):** We appreciate this suggestion to improve the timeliness of our paper. We have added four recent papers about network tomography published between 2020-2024 (citations [12], [19], [20], [21]). We have also included your recommended paper "Distributed

Network Tomography Applied to Stochastic Delay Profile Estimation" in our related work section.

[12] Jakub Kolar, Jan Sykora, Umberto Spagnolini, et al. Distributed network tomography applied to stochastic delay profile estimation. Radioengineering, 29(1):189–196, 2020

[19] Yuntong Hu and Liang Zhao. Deepnt: Path-centric graph neural networks for network tomography. 2024

[20] Ippokratis Sartzetakis and Emmanouel Varvarigos. Machine learning network tomography with partial topology knowledge and dynamic routing. In GLOBECOM 2022-2022 IEEE Global Communications Conference, pages 4922–4927. IEEE, 2022.

[21] Xu Tao and Simone Silvestri. Network tomography and reinforcement learning for efficient routing. In 2023 IEEE 20th International Conference on Mobile Ad Hoc and Smart Systems (MASS), pages 384–389. IEEE, 2023

3. **Regarding differences from previous work:** We have added a paragraph (lines 59-70) that clearly explains how this current work differs from and builds upon our previously published research.

However, these approach is ill-posed, as multiple sets of congested links (solutions) often match the observed congestion on the end-to-end paths, limiting its diagnostic performance. It should be noted that once a path traverses multiple congested links, its packet loss rate and communication delay are much higher than those of a single congested link. Based on this, this paper distinguishes and identifies the congestion status of paths by proposing a metric called Addictive Congestion Status and a method to measure it.

Comment 4:

Article - I may be wrong, but Fig. 1 is not cited in the paper text.

Response: Thank you for bringing this to our attention. We have added citations to Fig. 1 in line 50 and line 62 of the revised manuscript.

For instance, in **Figure 1(a)**, path y1 has a 9% packet loss rate traversing one congested link, while path y3 shows a 6% loss rate passing through two congested links. By leveraging this difference, we propose Additive Congestion Status A+ = R · X to quantify the number of congested links in each path. As illustrated in **Figure 1(b)**, this additional information helps reduce the solution space and improves diagnostic accuracy, achieving better precision and recall compared to Boolean approaches.

Comment 5:

Section 4 - please, check the labeling of "xi" in the text. It should be "x_i".

Response: Thank you for pointing out this notation issue. We have corrected the labeling from "xi" to "x_i" throughout the manuscript. The corrections can be seen in lines 245 and 246.

Specifically, $x_i=1$ indicates that the i-th link is congested, and $x_i=0$ indicates that the link is not congested

Comment 6:

Section 5 - information about the used HW/SW equipment and settings should be presented in detail. The authors made the mathematical model (source code) publicly available for reproducible research. It is perfect! However, the source code is not fully available. No details (e.g., preview of the source code is not visible). Please, check it!

Response: Thank you for this feedback. We have added detailed information about the hardware and software equipment used in our experiments in the setup section (lines 444-448). We have also improved the availability of our source code by uploading it to GitHub with more details and previews. The updated link is included in the paper.

Github link: https://github.com/Monickar/ACS

> The network event simulations were conducted with NS-3 (version 3.27) on a server equipped with Intel Platinum 8383C Official Edition CPU 40-Cores running Ubuntu 20.04 LTS. The duration of congestion in each simulation instance was set to 5 minutes. All experimental data and related code are available online.

Comment 7:

Section 5 - until Section 5.3, evaluation of the results is presented at a general level only. Their deeper discussion (for instance in terms of SOTA) is missing.

Response: Thank you for this suggestion. We have added a deeper discussion of our results in Section 5.2.4 (lines 570-581), particularly comparing our approach to state-of-the-art methods and analyzing the implications of our findings.

> Our findings revealed a notable improvement in recognizing ACS when utilizing the combined strength of LSTM and AAE, as opposed to employing either LSTM or AAE in isolation. The LSTM model demonstrated its ability to capture temporal dependencies within the network data effectively. In contrast, the AAE model excelled in learning a comprehensive representation of the data distribution. However, the combination of the LSTM and AAE significantly enhanced the framework's capability to classify the ACS accurately, demonstrating the complementary strengths of these models in handling the complexity and variability inherent in network tomography data.

Comment 8:

Section 5 - please, give more details about the used dataset. Next, please give more information about the testing - testing, training and validation - of the used (proposed) DL model.

Response: Thank you for requesting more details. We have added a comprehensive paragraph about our dataset (lines 468-482), including information about how it can be reproduced using our source code file (chinanet.cc). We have also expanded the information about our deep learning model's training, testing, and validation processes, including details about data splitting, handling class imbalance, and the hyperparameters used during training.

The dataset used in this research consists of end-to-end network measurements collected across multiple TopologyZoo network topologies (CHINANET, AGIS, GEANT, and ERNET). We systematically varied the congestion probability parameter (0.1-0.9) to create diverse network conditions, collecting path-level metrics including delay distributions, packet loss rates, and throughput measurements. The dataset was split into training (80\%) and testing (20\%) sets, with a validation subset used for hyperparameter tuning. To address class imbalance, we employed balanced sampling during model training. The LSTM-AAE model was trained using Adam optimizer (learning rate 0.001) with dropout regularization (0.5) and early stopping to prevent overfitting. Each scenario was repeated 40 times under different network conditions, resulting in approximately 3,000 samples per topology.

Comment 9:

Conclusion - information about future work plan is missing.

Response: Thank you for this observation. We have added our future work plans in the Conclusion section (lines 656-663), outlining several research directions we plan to explore, including extending our framework to handle dynamic network topologies, developing a lightweight version of our model, and incorporating transfer learning techniques.

> For future work, we plan to explore three main directions: (1) extending our framework to handle dynamic network topologies where routing paths change over time, (2) developing a lightweight version of our model to enable real-time congestion detection in resource-constrained environments, and (3) incorporating transfer learning techniques to adapt our pre-trained models to new network architectures with minimal retraining.

Reviewer #2

Comment 1:

As much as possible, do not separate figure from the description, like Figure 2.

Response: Thank you for this suggestion. We have adjusted the placement of all figures (Figures 2-9) to ensure they appear close to their corresponding description paragraphs in the text.

Comment 2:

If the formula is derived from a source, kindly cite it.

Response: Thank you for this reminder. We have reviewed our paper and found that the formulas in Section 4 are related to common metrics (F1, precision, recall) that are widely used in the field. These metrics do not have a specific source to cite as they are standard evaluation metrics. The formulas in Section 3 are our own derivations based on the network model we defined.

Comment 3:

Provide a detailed discussion of Figure 5 for clarity.

Response: Thank you for highlighting this need. We have enhanced the caption of Figure 5 and added a dedicated subsection (lines 410-431) that provides a detailed discussion of the figure. This new content explains our method's workflow, including how the measurement and inference phases operate, and clarifies the roles of different components in our proposed framework.

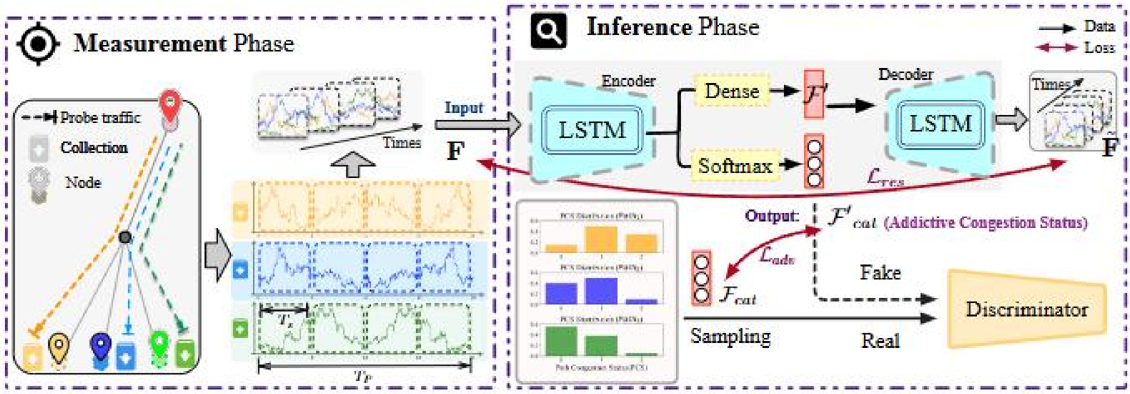


Figure 5: Overview of our proposed scheme. It has two phases: one is illustrated on the left side, for the process of end-to-end measurements and data collection; the other is depicted on the right side, for the ACS path status identification/inference via AAE-LSTM based deep learning. **Left:** The Measurement Phase collects probe traffic data over period TP with measurement windows Ts. **Right:** The Inference Phase processes this data through an encoder-decoder LSTM structure and a discriminator, outputting both reconstructed probe data (F) and Additive Congestion Status (F_cat), guided by loss functions L_res and L_adv .

Figure 5 illustrates our proposed two-phase framework for network congestion identification. In the measurement phase, probe traffic is sent through different network paths to collect path performance metrics over time T_P, with each probing action taking time T_s. The inference phase then processes this data through our deep learning architecture. Specifically, the collected probe information F is first fed into an LSTM encoder to capture temporal dependencies. The encoded features then pass through two parallel branches: one branch uses a dense layer followed by a decoder to reconstruct the input data for training stability, while the other branch employs a softmax layer to generate the Additive Congestion Status classification. In the adversarial training process, we sample from the true link congestion probability distribution to obtain real ACS labels, and train the discriminator to distinguish between these real samples and the predicted ACS from our model. This encourages the model to generate ACS predictions that match the underlying network congestion probability distribution, improving the accuracy of congestion status identification.

Comment 4:

Cite source about C-Link, Sum-Tomo and Netscope as competitor algorithms.

Response: Thank you for this important suggestion. We have added the necessary citations for these competitor algorithms in the experiment section (lines 591-592), properly acknowledging the original sources of CLINK, Sum-Tomo, and Netscope.

> Among the competitor algorithms, CLINK [7] only has the function of congested link diagnosis. Sum-Tomo [4] and Netscope [6] first infer the link performance based on the observation information of end-to-end paths and then diagnose congestion based on the inferred link performance.

Comment 5:

Emphasize the comparison of your proposed model (combination of LSTM and AAE) to LSTM and AAE only, where it can be found in your discussion. Ensuring that the comparison, the proposed model elevates the solution on the addressed gap from the study.

Response: Thank you for this suggestion. We have added content in the ablation study section (lines 570-575 and 581-585) to better highlight how our combined LSTM-AAE model addresses the research gaps identified in our study. The additional text explains how the temporal dependencies captured by LSTM and the distribution learning capabilities of AAE work synergistically to improve performance beyond what either approach could achieve alone.

To validate the efficacy of our framework combining AAE and LSTM networks, we conducted comparative experiments with four distinct models: LSTM only, LSTM \& AAE, and AAE only (a fully connected layer substituted the LSTM). These experiments were meticulously designed with identical experimental conditions and sample data to isolate the impact of each model configuration on the accuracy of ACS recognition.

Our findings revealed a notable improvement in recognizing ACS when utilizing the combined strength of LSTM and AAE, as opposed to employing either LSTM or AAE in isolation. The LSTM model demonstrated its ability to capture temporal dependencies within the network data effectively. In contrast, the AAE model excelled in learning a comprehensive representation of the data distribution. However, the combination of the LSTM and AAE significantly enhanced the framework's capability to classify the ACS accurately, demonstrating the complementary strengths of these models in handling the complexity and variability inherent in network tomography data.

Comment 6:

Specifically provide the names of traditional network tomography techniques to verify that the proposed model was compared from these traditional techniques in the methodology or in the experimental section of this study.

Response: Thank you for this suggestion. We have specifically named the traditional network tomography techniques that we compared our approach against in the conclusion section (lines 668-670). This clarifies that our experimental evaluation included direct comparisons with CLINK (Boolean tomography), Netscope (Analog tomography), and Sum-Tomo (Range tomography).

Specifically, our approach shows significant improvements over traditional network tomography techniques including CLINK (Boolean tomography), Netscope (Analog tomography), and Sum-Tomo (Range tomography), achieving better precision in congested link localization and more accurate link performance inference. By capturing the dynamic nature of network traffic, our AAE-LSTM framework provides a more robust solution for network congestion detection and quantification compared to these conventional approaches.

Comment 7:

Provide future work for researchers may consider in improving or enhancing the nature of your study.

Response: Thank you for this suggestion. We have added a discussion of future work in lines 656-663, outlining several research directions that could further improve and extend our approach. These include handling dynamic network topologies, developing a lightweight version of our model for resource-constrained environments, and incorporating transfer learning techniques for adapting pre-trained models to new network architectures.

For future work, we plan to explore three main directions: (1) extending our framework to handle dynamic network topologies where routing paths change over time, (2) developing a lightweight version of our model to enable real-time congestion detection in resource-constrained environments, and (3) incorporating transfer learning techniques to adapt our pre-trained models to new network architectures with minimal retraining.

Reviewer #3

Comment 1:

While the paper introduces ACS, it would be beneficial to explicitly highlight how ACS differs from traditional congestion detection metrics (e.g., Boolean tomography, Range tomography).

Response: Thank you for this suggestion. We have added two paragraphs (lines 46-70) in the introduction section that explicitly highlight how our proposed Additive Congestion Status (ACS) differs from traditional congestion detection metrics used in Boolean tomography and Range tomography. These paragraphs explain the limitations of traditional approaches and how ACS addresses these limitations by providing more granular path congestion information.

Based on a simple observation in normal network scenaries, a congested link will cause all paths passing through it to become congested. Therefore, network Boolean tomography extracts the link status by reversing the binary path status. Consider the example network shown in Figure 1, where paths {y1, y2, y3} traverse through five links {x1, x2, x3, x4, x5}. In traditional Boolean tomography, a congested link causes all paths passing through it to become congested, leading to binary path states B = R ∨ X.

However, this approach is ill-posed, as multiple sets of congested links (solutions) often match the observed congestion on the end-to-end paths, limiting its diagnostic performance. It should be noted that once a path traverses multiple congested links, its packet loss rate and communication delay are much higher than those of a single congested link. For instance, in Figure 1(a), path y1 has a 9% packet loss rate traversing one congested link, while path y3 shows a 6% loss rate passing through two congested links. By leveraging this difference, we propose Additive Congestion Status A+ = R · X to quantify the number of congested links in each path. As illustrated in Figure 1(b), this additional information helps reduce the solution space and improves diagnostic accuracy, achieving better precision and recall compared to Boolean approaches.

Comment 2:

A clearer motivation for using LSTM and AAE in combination should be provided. How does AAE contribute beyond just feature extraction?

Response: Thank you for this feedback. We have enhanced the motivation for combining LSTM and AAE in the introduction (lines 71-73). Additionally, we have added a new subsection 4.4 (lines 410-431) that explains the specific contributions of AAE beyond feature extraction. This section details how AAE regulates the latent space's data distribution to match the distribution of Additive Congestion Status information, which is crucial for accurate classification and quantification.

However, these approach is ill-posed, as multiple sets of congested links (solutions) often match the observed congestion on the end-to-end paths, limiting its diagnostic performance. It should be noted that once a path traverses multiple congested links, its packet loss rate and communication delay are much higher than those of a single congested link. Based on this, this paper distinguishes and identifies the congestion status of paths by proposing a metric called Addictive Congestion Status and a method to measure it. In detail, this approach integrates adversarial autoencoders with Long Short-Term Memory (LSTM) networks to refine the network tomography process. This hybrid approach leverages the adversarial autoencoder's ability to learn complex, non-linear representations of network paths and the LSTM's proficiency in capturing temporal dependencies within the path measurements.

Comment 3:

More details on the dataset (e.g., network size, traffic conditions) would improve the reproducibility of results.

Response: Thank you for this suggestion. We have added a comprehensive paragraph (lines 468-483) with detailed information about our dataset, including network sizes, traffic conditions, and congestion probability parameters. Additionally, we have noted that the dataset can be reproduced using our source code (chinanet.cc), which is publicly available on GitHub. The link to the repository is included in the paper for anyone wishing to replicate our results.

> The dataset used in this research consists of end-to-end network measurements collected across multiple TopologyZoo network topologies (CHINANET, AGIS, GEANT, and ERNET). We systematically varied the congestion probability parameter (0.1-0.9) to create diverse network conditions, collecting path-level metrics including delay distributions, packet loss rates, and throughput measurements. The dataset was split into training (80\%) and testing (20\%) sets, with a validation subset used for hyperparameter tuning. To address class imbalance, we employed balanced sampling during model training. The LSTM-AAE model was trained using Adam optimizer (learning rate 0.001) with dropout regularization (0.5) and early stopping to prevent overfitting. Each scenario was repeated 40 times under different network conditions, resulting in approximately 3,000 samples per topology.

Comment 4:

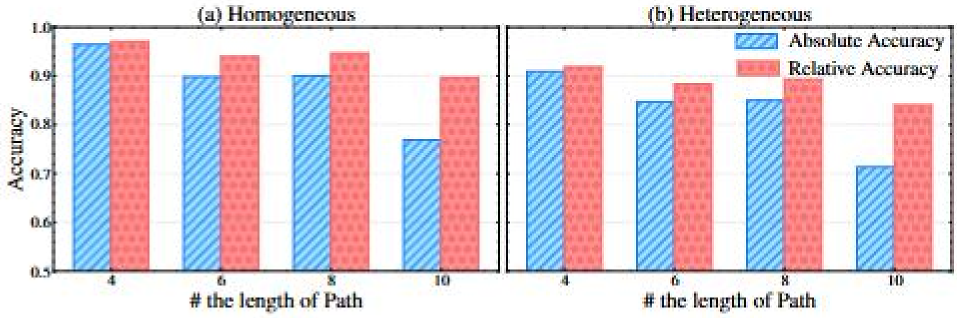
Were real-world datasets used, or is the evaluation purely based on simulated conditions? If simulated, how representative is it of actual network behavior?

Response: Thank you for this important question. Our evaluation is based on simulated conditions using NS-3 as the simulator. To make our simulations as representative as possible of real-world network behavior, we incorporated the PPBP (Poisson Pareto Burst Process) package (https://github.com/sharan-naribole/PPBP-ns3), which is based on the paper "A new tool for generating realistic Internet traffic in NS-3." While we acknowledge the limitations of not using real-world datasets, we have made significant efforts to ensure our simulated environments closely approximate actual network behavior by using realistic traffic patterns and topologies from TopologyZoo.

Comment 5:

Some figures (e.g., network congestion probability visualization) could be better labeled for clarity.

Response: Thank you for this feedback. We have improved the captions and labels of Figures 5, 7, 9, and 10 to enhance clarity. The revised captions provide more detailed explanations of the visualizations, making them more accessible to readers.



Estimation accuracy comparison between absolute path lengths (counting exact number of congested links) and relative path lengths (normalized distance from true value) from both homogeneous (a) and heterogeneous (b) network setups. The x-axis represents different path lengths ranging from 4 to 10 hops.

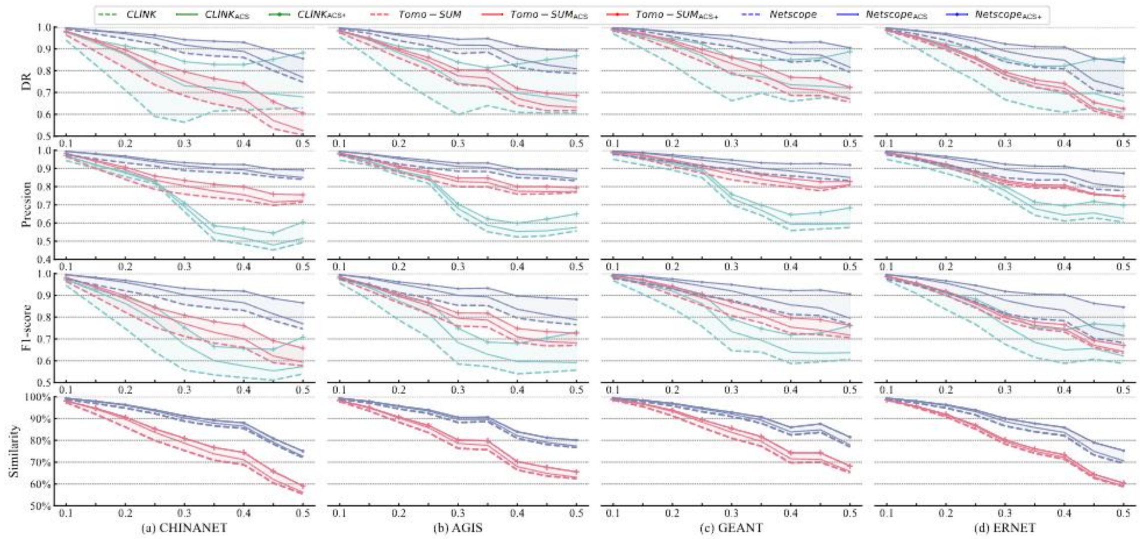


Figure 10: Performance comparisons among different network tomography algorithms under various link congestion probabilities. The subscript "ACS" and "ACS+" indicate qualitative (Aj ∈ {0, 1, 2}) and quantitative (A+ j ∈ {0, 1, 2, 3, ...}) ACS inputs respectively. Results are shown across four real network topologies: (a) CHINANET, (b) AGIS, (c) GEANT, and (d) ERNET.

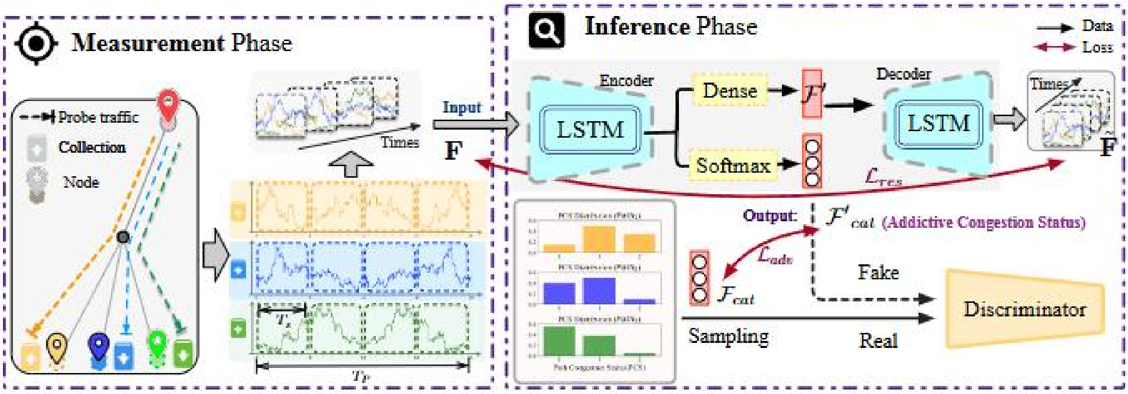


Figure 5: Overview of our proposed scheme. It has two phases: one is illustrated on the left side, for the process of end-to-end measurements and data collection; the other is depicted on the right side, for the ACS path status identification/inference via AAE-LSTM based deep learning. **Left:** The Measurement Phase collects probe traffic data over period TP with measurement windows Ts. **Right:** The Inference Phase processes this data through an encoder-decoder LSTM structure and a discriminator, outputting both reconstructed probe data (F) and Additive Congestion Status (F_cat), guided by loss functions L_res and L_adv .

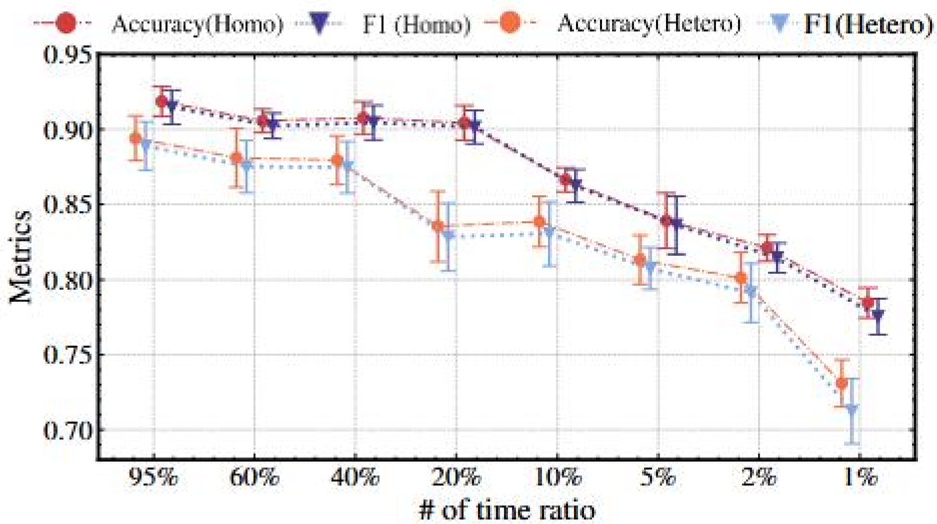


Figure 7: Accuracy and F1 scores of ACS-based congestion diagnosis under varying probing duration ratios. The experiments are conducted in both homogeneous (Homo) and heterogeneous (Hetero) network environments, showing the impact of observation time on model performance.

Comment 6:

The mathematical notation for ACS and inference processes should be briefly explained in the Introduction for readers unfamiliar with network tomography.

Response: Thank you for this useful suggestion. We have added explanations of the mathematical notation for ACS and the inference processes in the introduction section. This addition makes the paper more accessible to readers who may not be familiar with network tomography concepts.

Based on a simple observation in normal network scenarios, a congested link will cause all paths passing through it to become congested. Therefore, network Boolean tomography extracts the link status by reversing the binary path status through methods like Maximum-Likelihood estimation or greedy inference. Consider the example network shown in Figure 1, where paths {y1, y2, y3} traverse through five links {x1, x2, x3, x4, x5}. In traditional Boolean tomography, a congested link causes all paths passing through it to become congested, leading to binary path states $B = R \vee X$.

However, this approach is ill-posed, as multiple sets of congested links (solutions) often match the observed congestion on the end-to-end paths, limiting its diagnostic performance. It should be noted that once a path traverses multiple congested links, its packet loss rate and communication delay are much higher than those of a single congested link. For instance, in Figure 1(a), path y1 has a 9% packet loss rate traversing one congested link, while path y3 shows a 6% loss rate passing through two congested links. By leveraging this difference, we propose Additive Congestion Status $A+ = R \cdot X$ to quantify the number of congested links in each path. As illustrated in Figure 1(b), this additional information helps reduce the solution space and improves diagnostic accuracy, achieving better precision and recall compared to Boolean approaches.

Comment 7:

While CLINK, Netscope, and Sum-Tomo are useful baselines, it would strengthen the study if other machine learning-based approaches in network performance inference were discussed or compared.

Response: Thank you for this suggestion. We have added a discussion of machine learning and deep learning methods used for network tomography in line 139. We note that the essence of machine learning and deep learning methods for this problem is to fit an equation that solves for X (link status). Our ACS approach can be used to correct their judgment of X, typically requiring minor adjustments to the relevant algorithms. This correction reduces the distance d (defined in equation 6, with related proof in lines 295-321). We have incorporated this discussion to better position our work within the broader context of machine learning-based approaches to network tomography.

Method	Accuracy (%)		F1-score		NRMSE	
	w/o ACS	w/ ACS	w/o ACS	w/ ACS	w/o ACS	w/ ACS
DeepNT	81.2	86.5	0.79	0.85	0.31	0.15
NetTomo	79.8	82.6	0.77	0.81	0.35	0.18
PathInfer	80.3	84.0	0.78	0.83	0.33	0.16

We also expanded our evaluation to include recent machine learning approaches in network tomography. As shown in table, we tested our ACS framework with three representative deep learning methods: DeepNT (GNN-based), NetTomo (supervised learning), and PathInfer (GRU-based). The results demonstrate that incorporating ACS constraints consistently improves performance across all methods. Specifically, DeepNT's accuracy increased from 81.2% to 86.5%, while its NRMSE decreased from 0.31 to 0.15, showing significant improvement in both congestion detection and link performance inference. Similar enhancements were observed in NetTomo and PathInfer, with accuracy improvements of 2.8% and 3.7% respectively. These results validate our theoretical analysis that ACS helps reduce the solution space (as proven in equation 6), thereby improving the performance of various deep learning approaches in network tomography.

Comment 8:

Some sections contain minor grammatical errors and could benefit from professional proofreading.

Response: Thank you for pointing this out. We have carefully reviewed the manuscript and corrected grammatical errors throughout, including changing "majoreo" to "major" (line 94) and "two" to "three" (line 198). We have also performed a comprehensive proofreading to improve the overall language quality of the paper.

Comment 9:

The Related Work section could better differentiate how the proposed approach builds upon existing methodologies.

Response: Thank you for this suggestion. We have enhanced the Related Work section (lines 139-147) to more clearly differentiate how our proposed approach builds upon existing methodologies. This addition explains how our combined LSTM-AAE framework extends previous work and addresses limitations in traditional network tomography techniques.

Recently, machine learning approaches have shown promising results in network tomography. Several studies employ deep neural architectures like Graph Neural Networks to infer network structures or predict path performance. While these methods offer enhanced scalability and adaptability to diverse network conditions, they primarily focus on spatial dependencies in network topology, overlooking the temporal evolution of congestion patterns. Additionally, existing approaches often struggle with the high-dimensional solution space of congestion patterns, leading to potential misidentification of network status. Our work addresses these limitations through two key innovations: (1) incorporating LSTM networks to capture temporal dependencies in congestion evolution, and (2) leveraging adversarial autoencoders to learn the underlying distribution of congestion patterns, thereby reducing the solution space. This combined LSTM-AAE framework enables more accurate path congestion identification by jointly modeling both spatial and temporal characteristics of network behavior.