POSTER: DeePCAP: Enabling High-Fidelity and Cost-Efficient Archival Packet Trace Storage

Fenghao Dong
Carnegie Mellon University
Pittsburgh, PA, USA
fenghaod@andrew.cmu.edu

Yucheng Yin
Conviva
Foster City, CA, USA
sniperyyc@gmail.com

Yajie Zhou
University of Maryland
College Park, MD, USA
leszhou@umd.edu

Peilin Xin
Carnegie Mellon University
Pittsburgh, PA, USA
pxin@andrew.cmu.edu

Shinan Liu
University of Hong Kong
Hong Kong
shinan6@hku.hk

Vyas Sekar
Carnegie Mellon University
Pittsburgh, PA, USA
vsekar@andrew.cmu.edu

Abstract

Long-term network packet traces (e.g., pcaps), if available, can enable and inform lots of management tasks. However, storing packet data at scale is very expensive, forcing operators to choose between coarse historical summaries or short retention windows. In this context, deep generative compression (DGC) offers a new hope to store compact model parameters and regenerate structurally accurate traces on demand. We evaluate the suitability of recent deep generative approaches [21, 24] for packet trace modeling and generation. We find that their fidelity metrics are disconnected from the domain-specific queries/use cases, and they have bad cost-fidelity trade-off. We propose DeePCAP, an end-to-end trace storage system to close this gap. DeePCAP introduces a query-driven fidelity framework spanning packet- and flow-level queries to tackle the fidelity disconnection, and proposes a novel dimensionality reduction approach using frequency domain encoding to improve cost-fidelity trade-off. Our preliminary results show that DeePCAP achieves the best fidelity on the 100+ query suite and the strongest cost–fidelity trade-off.

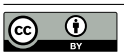
CCS Concepts

• **Networks** → **Network measurement**; • **Information systems** → *Information storage systems*; • **Computing methodologies** → *Machine learning*.

Keywords

packet traces, trace storage, deep generative compression, network measurement, data compression, frequency-domain encoding

ACM Reference Format:
Fenghao Dong, Yucheng Yin, Yajie Zhou, Peilin Xin, Shinan Liu, and Vyas Sekar. 2026. POSTER: DeePCAP: Enabling High-Fidelity and Cost-Efficient Archival Packet Trace Storage. In *ACM SIGCOMM Posters and Demos (SIGCOMM Posters and Demos '26), August 17–21, 2026, Denver, CO, USA.* ACM, New York, NY, USA, 3 pages. https://doi.org/10.1145/3789240.3830275



This work is licensed under a Creative Commons Attribution 4.0 International License.
SIGCOMM '26, Denver, CO, USA
© 2026 Copyright held by the owner/author(s).
ACM ISBN 979-8-4007-2467-1/26/08
https://doi.org/10.1145/3789240.3830275

1 Background and Motivation

Network packet header traces play a critical role in enabling security operators to conduct a wide range of longitudinal security-related tasks. These tasks include, but are not limited to, traffic analysis for network forecasting and monitoring [14, 15, 20], testing and evaluating defense systems to ensure their robustness and effectiveness [2, 7, 19], and training machine learning models for intrusion detection, traffic classification, and adversarial resilience [9, 11, 18, 22, 28]. Given the increasing complexity and volume of modern network environments, operators would ideally like to store longitudinal packet traces over extended time periods to inform and power such management functions.

One major challenge with storing raw packet header traces is the cost associated with storing longitudinal packet traces. The high volume [12, 15] of packet-level data generated by modern networks makes long-term storage expensive in terms of infrastructure and maintenance. This creates a trade-off for operators between retaining critical data for analysis and meeting budget obligations. Even applying traditional compression [16] or state-of-the-art compression algorithms [1, 6, 8, 13] can result in significant storage costs at scale. This forces operators into either using only a small retention window of recent header capture data for analysis or only retain coarse statistics over history.

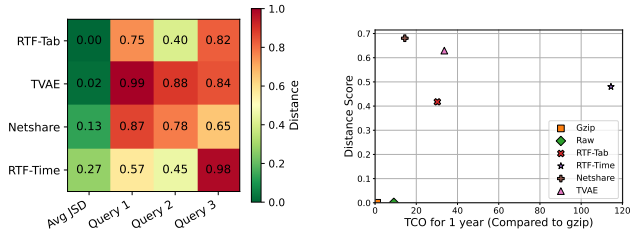
In this light, Deep Generative Compression (DGC) offers a promising solution. The observation is that most of the aforementioned management use cases do not require exact replay of packet traces but require *structurally accurate reconstructions* [3, 4, 10, 24]. That is, these types of tasks are not "forensic" tasks looking for a specific packet or flow but are analyzing structural properties of the traffic as a whole and looking for patterns. By using a deep generative model of historical trace snapshots and reconstructing them on demand (e.g., NetShare [24], FlowChronicle [5], NetDiff [26]), DGC has the potential to reduce storage costs by up to 99.8% compared to state-of-the-art compression schemes.

Inspired by this promise, we evaluate the suitability of proposals for using deep generative models for packet traces (e.g., [21, 23–25, 27, 29]). Unfortunately, we see that current approaches have two key limitations.

Limitation 1: Poor fidelity on network-centric queries (fig 1a). Despite having good-looking fidelity on ML-centric metrics used by prior works like average Jensen–Shannon divergence averaged over columns, when we run them on three actual network-centric

queries (like Identify which destination IPs are most frequently contacted through the busiest destination port each day), we found that they actually have very poor fidelity on them.

Limitation 2: Poor cost-fidelity trade-off (fig 1b). Despite having impressive capabilities to reduce the storage cost, if we look into total cost of ownership (TCO), which includes both the model training cost and storage cost, DGC is not cost-efficient at all. It has much worse cost-fidelity trade-off compared to even storing raw packets.



(a) Mismatch between average JSD and network-centric use case fidelity.

(b) Poor Cost-fidelity trade-off compared to traditional lossless compression.

Figure 1: Limitations of existing DGC approaches.

Goal. Our goal is to build DeePCAP, a practical compression system for network packet headers which can achieve both high fidelity on network-centric tasks, and competitive cost-fidelity trade-off compared to traditional lossless compression schema.

2 System Design

We start from first principles to understand why existing work falls short. We find two key sources of misalignment in their design choices:

- *Fidelity misalignment:* On the fidelity dimension, we observe that existing models only use ML-centric metrics (e.g., per-field Jensen Shannon distance) during training process. But from fig 1a we can see they are disconnected from the fidelity on network-centric tasks. Therefore, although those models perform well in term of ML-centric metrics, they perform bad on actual network tasks. To solve this problem, we started from various network-centric tasks, and design a comprehensive set of over 100 queries which captures packet-level and flow-level features that are interesting to domain experts and are frequently used in downstream analytic tasks. Then we use this set of queries for model selection and hyper-parameter fine-tuning during training process.
- *Performance misalignment:* Prior work either modeling *all packets together as a table* or *all flows together as a packet arrival time-series*. Both approaches have key shortcomings. While tabular models are simpler and more stable, modeling packets as a table loses key flow-level and stateful properties. On the other hand, time-series models are harder to learn and suffer a dimensionality problem bottle-necked by the longest flows. In fact, this training time even perversely impacts simple aggregate queries of interest. To solve this problem, we move from viewing packets time-series in time-domain to a more compact *frequency-domain* representation. Specifically, we treat each measurement column (i.e. the columns except 5-tuple) as a signal. Then we perform sparse approximation using OMP (orthogonal matching pursuit) [17] to encode the measurement signals in frequency domain, then we can use simpler and more stable tabular models over the frequency domain coefficients.

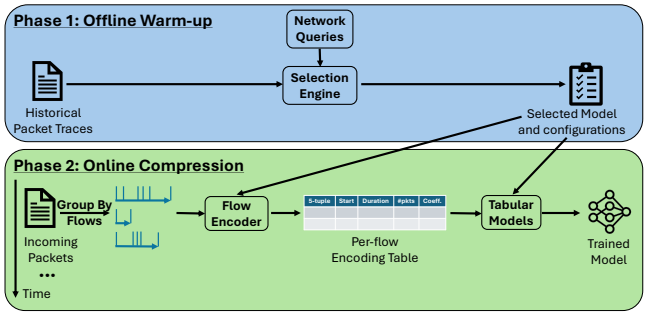
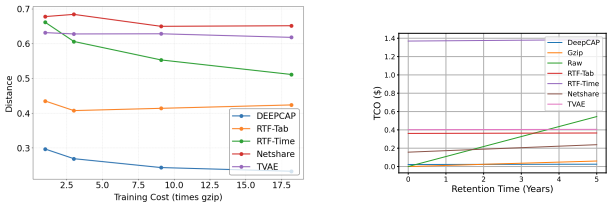


Figure 2: End-to-end design of DeePCAP.

The end-to-end design of DeePCAP is shown in figure 2. During offline warm-up phase, given sample packet traces collected in the past, the selection engine tries to select the model and configuration (like model hyperparameter, bases for frequency domain encoding, etc) based on the fidelity on our set of network-centric queries. In online compression phase, we divide incoming packets to measurement epochs, and for each epoch, first we group them by 5-tuple, then encode them in frequency-domain using the bases selected during offline phase and yield the per-flow encoding table. Afterwards, we train the selected tabular model using the selected hyperparameter on it, and store the trained model parameters. tune on the incoming streaming data and compress the traces periodically.

3 Preliminary Results

We evaluated DeePCAP and baselines on M57 dataset, which contains 5.4M packets in a span of 28 days. Figure 3a shows that DeePCAP has better cost-fidelity trade-off compared to other models. Figure 3b shows that DeePCAP has lower TCO than gzip with over 2 years of retention, while other models need over 20 years to be cost competitive to gzip.



(a) DeePCAP has better Cost-fidelity trade-off than other models.

(b) DeePCAP is cheaper than gzip with over 2 years of retention.

Figure 3: Cost and fidelity evaluation.

References

[1] Azim Afroozeh, Leonardo X Kuffo, and Peter Boncz. 2023. Alp: Adaptive lossless floating-point compression. *Proceedings of the ACM on Management of Data* 1, 4 (2023), 1–26.

[2] Mohammad Salim Ahmed, Ehab Al-Shaer, and Latifur Khan. 2008. A novel quantitative approach for measuring network security. In *IEEE INFOCOM 2008-The 27th Conference on Computer Communications*. IEEE, 1957–1965.

[3] Andrew Chu, Xi Jiang, Shinan Liu, Arjun Bhagoji, Francesco Bronzino, Paul Schmitt, and Nick Feamster. 2024. Feasibility of state space models for network traffic generation. In *Proceedings of the 2024 SIGCOMM Workshop on Networks for AI Computing*. 9–17.

[4] Andrew Chu, Xi Jiang, Shinan Liu, Arjun Bhagoji, Francesco Bronzino, Paul Schmitt, and Nick Feamster. 2025. NetSSM: Multi-Flow and State-Aware Network Trace Generation using State-Space Models. *arXiv preprint arXiv:2503.22663* (2025).

[5] Joscha Cüppers, Adrien Schoen, Gregory Blanc, and Pierre-Francois Gimenez. 2024. Flowchronicle: Synthetic network flow generation through pattern set mining. *Proceedings of the ACM on Networking* 2, CoNEXT4 (2024), 1–20.

[6] Google. 2026. Snappy: A fast compressor/decompressor. https://github.com/google/snappy.

[7] Ragini Gupta, Shinan Liu, Ruixiao Zhang, Xinyue Hu, Xiaoyang Wang, Hadjer Benkraouda, Pranav Kommaraju, Phuong Cao, Nick Feamster, and Klara Nahrstedt. 2025. Generative active adaptation for drifting and imbalanced network intrusion detection. *arXiv preprint arXiv:2503.03022* (2025).

[8] Kaisei Hishida, Chunwei Liu, John Paparrizos, and Aaron J Elmore. 2025. Beyond Compression: A Comprehensive Evaluation of Lossless Floating-Point Compression. *Proceedings of the VLDB Endowment* 18, 11 (2025), 4396–4409.

[9] Arthur S Jacobs, Roman Beltiukov, Walter Willinger, Ronaldo A Ferreira, Arpit Gupta, and Lisandro Z Granville. 2022. AI/ML for network security: The emperor has no clothes. In *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security*. 1537–1551.

[10] Xi Jiang, Shinan Liu, Aaron Gember-Jacobson, Arjun Nitin Bhagoji, Paul Schmitt, Francesco Bronzino, and Nick Feamster. 2024. Netdiffusion: Network data augmentation through protocol-constrained traffic generation. *Proceedings of the ACM on Measurement and Analysis of Computing Systems* 8, 1 (2024), 1–32.

[11] Muskan Khan and Laiba Ghafoor. 2024. Adversarial Machine Learning in the Context of Network Security: Challenges and Solutions. *Journal of Computational Intelligence and Robotics* 4, 1 (2024), 51–63.

[12] Franck Le, Mudhakar Srivatsa, Raghu Ganti, and Vyas Sekar. 2022. Rethinking data-driven networking with foundation models: challenges and opportunities. In *Proceedings of the 21st ACM Workshop on Hot Topics in Networks*. 188–197.

[13] Chunwei Liu, Hao Jiang, John Paparrizos, and Aaron J Elmore. 2021. Decomposed bounded floats for fast compression and queries. *Proceedings of the VLDB Endowment* 14, 11 (2021), 2586–2598.

[14] Shinan Liu, Francesco Bronzino, Paul Schmitt, Arjun Nitin Bhagoji, Nick Feamster, Hector Garcia Crespo, Timothy Coyle, and Brian Ward. 2023. LEAF: Navigating Concept Drift in Cellular Networks. *Proceedings of the ACM on Networking* 1, CoNEXT2 (2023), 1–24.

[15] Shinan Liu, Ted Shaowang, Gerry Wan, Jeewon Chae, Jonatas Marques, Sanjay Krishnan, and Nick Feamster. 2024. ServeFlow: A Fast-Slow Model Architecture for Network Traffic Analysis. *arXiv preprint arXiv:2402.03694* (2024).

[16] Jean loup Gailly and Mark Adler. 1996. gzip home page. https://www.gzip.org/. Accessed: 2024-12-22.

[17] Yagyensh Chandra Pati, Ramin Rezaiifar, and Perinkulam Sambamurthy Krishnaprasad. 1993. Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition. In *Proceedings of 27th Asilomar conference on signals, systems and computers*. IEEE, 40–44.

[18] Kamran Shaukat, Suhuai Luo, Vijay Varadharajan, Ibrahim A Hameed, and Min Xu. 2020. A survey on machine learning techniques for cyber security in the last decade. *IEEE access* 8 (2020), 222310–222354.

[19] Hadi Shiravi, Ali Shiravi, and Ali A Ghorbani. 2011. A survey of visualization systems for network security. *IEEE Transactions on visualization and computer graphics* 18, 8 (2011), 1313–1329.

[20] Leslie F Sikos. 2020. Packet analysis for network forensics: A comprehensive survey. *Forensic Science International: Digital Investigation* 32 (2020), 200892.

[21] Aivin V Solatorio and Olivier Dupriez. 2023. Realtabformer: Generating realistic relational and tabular data using transformers. *arXiv preprint arXiv:2302.02041* (2023).

[22] Gerry Wan, Shinan Liu, Francesco Bronzino, Nick Feamster, and Zakir Durumeric. 2025. CATO: End-to-End Optimization of ML-Based Traffic Analysis Pipelines. In *22nd USENIX Symposium on Networked Systems Design and Implementation (NSDI 25)*. 1523–1540.

[23] Lei Xu, Maria Skoularidou, Alfredo Cuesta-Infante, and Kalyan Veeramachaneni. 2019. Modeling tabular data using conditional gan. *Advances in neural information processing systems* 32 (2019).

[24] Yucheng Yin, Zinan Lin, Minhao Jin, Giulia Fanti, and Vyas Sekar. 2022. Practical gan-based synthetic ip header trace generation using netshare. In *Proceedings of the ACM SIGCOMM 2022 Conference*. 458–472.

[25] Hengrui Zhang, Jiani Zhang, Balasubramaniam Srinivasan, Zhengyuan Shen, Xiao Qin, Christos Faloutsos, Huzefa Rangwala, and George Karypis. 2024. Mixed-Type Tabular Data Synthesis with Score-based Diffusion in Latent Space. In *The twelfth International Conference on Learning Representations*.

[26] Shiyuan Zhang, Tong Li, Depeng Jin, and Yong Li. 2024. NetDiff: a service-guided hierarchical diffusion model for network flow trace generation. *Proceedings of the ACM on Networking* 2, CoNEXT3 (2024), 1–21.

[27] Yajie Zhou, Jason Li, Gianluca Stringhini, Ayse K Coskun, and Zaoxing Liu. 2023. Enabling Privacy-preserving Multidimensional Network Telemetry with Autoencoders. In *2023 IEEE International Conference on Cloud Engineering (IC2E)*. IEEE, 228–229.

[28] Yajie Zhou, Jiajun Ruan, Eric S Wang, Sadjad Fouladi, Francis Y Yan, Kevin Hsieh, and Zaoxing Liu. 2026. NetArena: Dynamic Benchmarks for AI Agents in Network Automation. In *The Fourteenth International Conference on Learning Representations*.

[29] Yajie Zhou, Fuheng Zhao, Eric Wang, Ayse K. Coskun, Divyakant Agrawal, Amr El Abbadi, and Zaoxing Liu. 2026. PrvTel: Lightweight Models for Private and Accurate Telemetry Data Retention. In *23rd USENIX Symposium on Networked Systems Design and Implementation (NSDI 26)*. USENIX Association, Renton, WA, 1827–1843.