GRAPH LEARNING FOR REAL-WORLD APPLICATIONS

a research proposal prepared by QIMING GUO
October, 2025

for
The Graduate Committee
Computer Science Program
College of Engineering and Computer Science
Texas A&M University–Corpus Christi
Corpus Christi, Texas

Approved:

Dr. Wenlu Wang, Chairperson

Dr. Taoran Ji, Member

Dr. Zijie Zhang, Member

Dr. Carlos Rubio Medrano, Member

Abstract

In recent years, AI has advanced rapidly and gradually produced positive impacts in research and applications across various disciplines. Graph learning is one of the most promising directions in AI research and application. In this research proposal, I explore how graph learning can be used to build an automated AI system, with the goal of benefiting both human interests and the environment. To achieve this goal, I propose three staged objectives. The first objective is graph neural networks for real-world applications. The second objective is trustworthy and secure AI. The third objective is to deploy AI to real world.

Currently, I have completed part of the work. For the first objective, as a representative case study in wastewater monitoring, I have modeled a spatiotemporal graph, demonstrated data usability and predictability, and constructed graph-structured data representations. For the second objective, I developed HydroNet, a domain-specific ST-GNN that incorporates relevant features, achieving state-of-the-art results on real datasets from the case study. For the third objective, I explored partition-and-merge strategies, enabling local retraining of subgraphs instead of the entire graph. Initial results show reduced costs but some accuracy loss. In this research proposal, I also present plans for further advancing each of these objectives.

Contents

1 Introduction

1.1 Motivation & Problem Statement

Graph learning represents a transformative approach in artificial intelligence, facilitating the analysis and prediction of interconnected data structures prevalent in numerous real-world scenarios. From transportation and energy networks to environmental monitoring and social systems, graphs naturally capture relationships and dependencies that traditional machine learning methods often overlook. Despite its potential, deploying graph learning in practical applications encounters significant hurdles that must be addressed to realize its full benefits.

The primary challenge lies in data acquisition and preparation. Real-world data for graph-based models must be not only abundant but also high-quality, featuring spatiotemporal regularities that enable effective representation and learning. This involves careful feature engineering, data collection methodologies, quality assurance, availability assessment, and the construction of graph-structured formats. For instance, in a wastewater monitoring system as a case study, sensors must be strategically placed to capture flow dynamics, but similar considerations apply to traffic sensors in urban mobility or patient interactions in healthcare networks. Without robust data pipelines, graph models cannot achieve reliable performance, underscoring the need for systematic approaches to data usability and predictability.

Once suitable data is secured, the next obstacle is model adaptation. Generic graph neural networks (GNNs), while versatile, frequently underperform in domain-specific contexts because they fail to incorporate unique characteristics such as physical laws, directional flows, or temporal periodicity. Designing specialized GNNs that integrate these priors is essential for enhancing accuracy and interpretability. Continuing with the wastewater example, models must account for pipe attributes and hydraulic principles, but analogous adaptations are required in other fields, like incorporating traffic rules in transportation forecasting or biological constraints in molecular modeling.

Finally, scalability and maintenance pose ongoing challenges. As graphs grow—due to expanding networks or accumulating data—updates to topology or content (e.g., node additions/removals for privacy compliance) can necessitate costly retraining. In large-scale systems, even minor changes, such as altering a few nodes, can demand full model recomputation, leading to inefficiencies in time, computation, and energy. Privacy concerns further complicate this, as data deletion requests must be honored without compromising overall system integrity. These issues are universal across applications, from dynamic urban infrastructures to evolving social graphs, demanding innovative strategies for efficient, secure, and scalable graph learning.

Addressing these interconnected challenges is vital for transitioning graph learning from theoretical advancements to impactful real-world deployments, ultimately benefiting diverse sectors

by enabling automated, intelligent systems.

1.2 Research Objectives

Graphs have their unique advantages; they are very suitable for applications in the real world, especially in scenarios with graph structures and temporal attributes. However, applying Graphs in real-world modeling and prediction is challenging. This research aims to advance graph learning for real-world applications, using wastewater monitoring as a representative case study to demonstrate practical impacts. To achieve this, the following objectives guide the work:

1. **Objective 1: Graph Neural Network for Real World Applications.**

   Motivation: Graph learning can automate monitoring and prediction in interconnected systems, relieving humans from inefficient tasks in real-world domains like infrastructure networks.

   Progress: As a case study, I have modeled a wastewater system as a spatiotemporal graph, demonstrated data usability and predictability, and constructed graph-structured data representations.

   Next Steps: Build integrated AI systems that automate monitoring, prediction, and reporting—combining real-time ST Graph forecasting with large language models for interpretable outputs across applications.

2. **Objective 2: Trustworthy and Secure AI.**

   Motivation: Generic ST-GNNs often fail to incorporate domain-specific properties, limiting trustworthiness and security in real-world data scenarios.

   Progress: I developed **HydroNet**, a domain-specific ST-GNN that incorporates relevant features, achieving state-of-the-art results on real datasets from wastewater systems as a case.

   Next Steps: Extend models with richer priors (e.g., physical attributes) and explore ST-MPNN architectures for enhanced accuracy, interpretability, and security.

3. **Objective 3: Deploy AI to Real World.**

   Motivation: Scalability, privacy, and efficiency are critical for deploying graph learning in dynamic real-world systems, where updates or deletions require minimal resources.

   Progress: I explored partition-and-merge strategies, enabling local retraining of subgraphs with reduced costs and minor accuracy loss. Using global compensation (e.g., MLP), accuracy improved from 90% to 95%.

Next Steps: Develop frameworks for near-lossless scalability and privacy-conscious unlearning through advanced integration, ensuring practical deployment.

1.3 Relevance and Research Necessity

Graph learning holds immense relevance for addressing complex problems in interconnected real-world systems, where traditional methods fall short in capturing relational dynamics. In domains like urban infrastructure, energy distribution, healthcare, and environmental management, graphs provide a natural framework for modeling entities and their interactions, enabling predictions, optimizations, and anomaly detections that drive efficiency and sustainability. As global challenges—such as resource scarcity, climate change, and data privacy—intensify, the necessity for robust graph-based AI becomes evident, offering tools to manage vast, evolving networks with precision.

However, significant gaps persist in bridging theoretical graph learning with practical deployment. First, real-world data often suffers from sparsity, noise, or incompleteness, hindering effective graph construction and learning (aligning with Objective 1). Second, generic models lack trustworthiness, as they may ignore domain-specific knowledge, leading to unreliable outputs or vulnerabilities (Objective 2). Third, deployment in dynamic environments requires security features like privacy-preserving unlearning and scalability, which current approaches inadequately address (Objective 3). For example, in wastewater systems as a case, sparse sensing and privacy regulations exemplify these issues, but they mirror broader challenges in traffic networks or social platforms.

Recent advances in graph neural networks (GNNs), spatio-temporal extensions (ST-GNNs), message-passing neural networks (MPNNs), and unlearning techniques lay a strong foundation. Spectral methods enhance partitioning, transformers improve long-range forecasting, and privacy-focused frameworks like differential privacy ensure compliance. Despite these, innovations are needed for sparse observability, physics-informed modeling, and secure scalability to make graph learning practically viable.

Therefore, this research is essential to close the divide between graph learning theory and real-world necessities. By tackling data modeling, trustworthy adaptations, and secure deployment, it will foster AI systems that are not only accurate and efficient but also ethically sound and broadly applicable across critical domains.

2 Background and Related Work

2.1 Graph Neural Networks and Spatio-Temporal Extensions

Graphs are fundamental structures for modeling relational data, comprising a set of nodes $\mathcal{V}$ and edges $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$, represented by an adjacency matrix $A \in \mathbb{R}^{N \times N}$ where $A_{uv} > 0$ indicates a connection. The Laplacian matrix $L = D - A$, with degree matrix D, enables spectral analysis for tasks like clustering. Spatio-temporal graphs extend this by integrating spatial adjacency and temporal dynamics, defined as $\mathcal{G} = (\mathcal{V}, \mathcal{E}, X)$ where $X \in \mathbb{R}^{T \times N \times F}$ captures time-series features over T timesteps with F dimensions (e.g., traffic speed). Each node $v \in \mathcal{V}$ has spatial coordinates $\mathbf{s}_v$ (e.g., geographic locations) and temporal states $\mathbf{x}_v^t$, reflecting dynamics like periodic traffic patterns or event-driven spikes Yu et al. (2018a); Li et al. (2018). Applications span traffic forecasting, epidemiological modeling Kapoor et al. (2020), and healthcare. Recent advances, such as Graph Convolutional Networks (GCN) Kipf and Welling (2016), Graph Attention Networks (GAT) Veličković et al. (2018), and Spatio-Temporal Graph Convolutional Networks (ST-GCN) Yu et al. (2018a), leverage multi-hop propagation and temporal dependencies to predict future states $\hat{y}v$ via $\Theta^* = \arg\min \Theta \mathcal{L}(\Theta; \mathcal{D})$ Wu et al. (2019b), enhancing accuracy but complicating unlearning due to global couplings.

2.1.1 Modeling Hydraulic Systems with Graphs

Traditional physic-based models, based on hydraulic principles, are computationally demanding and often unsuitable for real-time applications or large scale simulations. They require extensive calibration and are limited in handling dynamic or sparse data Zhang (2024). Conversely, GNNs offer a data driven alternative that is well suited to networked systems utilizing both topological structure and physical attributes for efficient hydraulic modeling.

In water distribution systems (WDS), GNNs are adopted for reconstructing nodal pressures Hajgató et al. (2021). With an edge-weighting mechanism, this approach exhibits robust nodal pressure recovery performance, even with limited instrumentation. GNNs are also utilized to integrate water network partitioning and dynamic district-metered areas Rong et al. (2021). Notably, besides supervised learning, semi-supervised learning is also explored for state estimation in WDS using graph neural networks Xing and Sela (2022). GNN-based approaches have showcased effectiveness in optimizing WDS management and responding to emergencies.

Another line of research combines GNN with recurrent processing units to capture spatial and time-series temporal relations. For example, a Graph Convolutional Recurrent Neural Network (GCRNN) Zanfei et al. (2022) is proposed for water demand forecasting, outperforming traditional LSTM models, especially during sensor malfunction conditions. Additionally, Graph Neu-

ral Rainfall-Runoff Model (GNRRM) Xiang and Demir (2021) is another GNN based model for high-resolution rainfall-runoff modeling. It enables the use of spatial relations from high-resolution precipitation data, including flow direction and geographic features.

A specialized ST-GNN, Graph WaveNet Wu et al. (2019a), utilizes dilation to condense time-series data before passing it to a GCN, enabling the capture of spatial-temporal dependencies. Its effectiveness has been demonstrated in precise groundwater level forecasting Bai and Tahmasebi (2023) and predicting streamflow in basin-scale river networks Sun et al. (2022). ST-GNNs have demonstrated superior performance in hydraulic modeling compared to traditional methods. However, many of these models fail to leverage critical graph features, such as edge information.

2.1.2 Message Passing Neural Networks

MPNNs have been employed in computational chemistry to predict the molecular properties Gilmer et al. (2017); St John et al. (2019); Jo et al. (2022). Gilmer et al. (2017) formulated a common MPNN framework for learning in graph structured data. It consists of two phases: message passing phase over T time steps and a readout phase to generate the prediction. During message passing, each node v updates its hidden state h_v^t by aggregating the messages from its neighbors:

$$m_v^{t+1} = \sum_{w \in \mathcal{N}(v)} M_t(h_v^t, h_w^t, e_{vw})$$

$$h_v^{t+1} = U_t(h_v^t, m_v^{t+1})$$

where M_t is the message function, U_t is the update function and e_{vw} denotes the edge features. After T iterations, a readout function R computes the graph level output:

$$\hat{y} = R\left(\{h_v^T \mid v \in \mathcal{G}\}\right)$$

M_t, U_t and R are learnable differential functions.

The authors applied this framework to predict quantum chemical properties of molecules in the QM9 Ramakrishnan et al. (2014) dataset. They explored several novel MPNN variants, such as different message functions, virtual edges, different readout functions, achieving state-of-the-art results, suggesting it as a flexible and powerful tool Gilmer et al. (2017). Similarly, a duel branched MPNN architecture combining standard MPNN and a separate MLP branch for molecular property prediction was proposed Jo et al. (2022). The MPNN captured the interatomic interaction using edge features: distance and angle while MLP modeled the single-atom features to mitigate over smoothing. Likewise, St John et al. (2019) also implemented MPNN to predict the optoelectronic properties of over 91,000 organic molecules using only 2D molecular structures. The results

showed that the MPNN, trained on 2D structures with discrete nodes (atoms) and edge (bond) features such as bond type, conjugation and atom pairs, demonstrated similar accuracy to that of models that utilized 3D geometric, thereby significantly reducing the computational cost. The edge features were encoded as learned weight metrics to modulate the message passing process between the atoms.

These studies highlight importance of edge features, especially in chemistry, where they significantly affect the molecular behavior. The result showed that MPNNs are well suited for this domain as they allow integration of such edge information into the learning process. This concept can be extended to water and wastewater (sanitary sewer) systems, where the hydraulic behavior between the manholes is influenced by not only node properties but also the characteristics of connecting pipe. Features like pipe length, slope, material, diameter determine how the flow propagates in the system. Therefore, by enabling the inclusion of pipe features, MPNNs offer a powerful and flexible approach to capture the hydraulic dynamics in the sewer system.

2.1.3 Spatio-Temporal Prediction: Adaptive Graphs, Transformers, and MoE

Recent advances since 2019 have shifted pipeline anomaly detection from recurrent models toward graph- and attention-based architectures. Graph WaveNet Wu et al. (2019b) and its extensions Cui et al. (2020); Fang et al. (2021) demonstrated the effectiveness of adaptive graph learning for irregular network topologies, while transformer-based methods Zeng et al. (2021); Wu et al. (2021); Li et al. (2023) enabled long-range spatiotemporal forecasting under sparse or noisy sensing. More recently, mixture-of-experts frameworks Lepikhin et al. (2020); Zhai et al. (2023) have further improved robustness and efficiency by combining specialized predictors. These developments provide the foundation for systems like AquaSentinel, where reliable and scalable spatiotemporal prediction is essential for real-time anomaly detection.

2.2 Case Study: Challenges and Existing Methods for Real-World Applications

2.2.1 Recent Major Incidents and Motivation

Recent major leak incidents in the United States highlight these issues: in Houston, Texas, a 96-inch water main break prompted a city-wide boil-water advisory and flooded streets and a major freeway segment cas (2020); in Dallas, Texas, a leak in a treated water pipeline led to an estimated 3.6 million gallons lost, creating swamp-like conditions in a forested area cas (2023); in Tuscaloosa, Alabama, a 275,000-gallon sewer line leak triggered emergency repairs cas (2025a); in Atlanta, Georgia, water main issues led to boil advisories in affected neighborhoods, including

Buckhead, disrupting homes and businesses cas (2024a); in Richmond, Virginia, a winter-storm-related power outage at the water treatment plant caused widespread service loss and a boil-water advisory across the region cas (2025b); in Washington, DC, water main breaks shut down a section of Martin Luther King Jr. Avenue SE, closing roads and disrupting local service cas (2025c); in Carnelian Bay, California, a sewer spill of tens of thousands of gallons closed beaches, prompted health advisories, and led to an $850,000 settlement cas (2024b); and in Detroit, Michigan, a major water main break affected nearly 400 homes, requiring evacuations and emergency sheltering while causing property and infrastructure disruption cas (2025d). Furthermore, some pipeline anomalies manifest as background leakage—small, diffuse leaks occurring at joints or minor structural defects, which typically remain hidden, undetected by instruments, and unnoticed at the surface until they have polluted the environment, wasted water resources, or gradually undermined adjacent structural integrity. Other incidents are visible leakage, where the leakage becomes apparent to observers. Crucially, many severe visible leaks have their origins in background leakage that has progressively worsened. These cases demonstrate how undetected or slowly addressed leaks can result in widespread service disruptions, environmental contamination, and significant financial burdens, underscoring the urgent need for more accessible and cost-effective detection solutions.

a. Background Leakage b. Visible Leakage

Figure 1: Critical pipeline infrastructure failures demonstrating the urgent need for automated monitoring systems. **(a) Background Leakage:** The silent majority of pipeline failures occur as gradual, undetected leaks that persist for months or years. These images show typical background leakage scenarios including subsurface water accumulation, soil erosion, corroded pipe segments, and contaminated groundwater infiltration. Such leaks waste millions of gallons of clean water annually and allow untreated sewage to contaminate soil and groundwater, often going unnoticed until environmental damage becomes irreversible. **(b) Visible Leakage:** Catastrophic failures that disrupt urban life, resulting from either sudden mechanical damage (construction accidents, ground movement) or progressive deterioration from undetected background leaks that escalate to structural collapse.

2.2.2 Manual and Sensor-Based Detection Methods

Traditional leak detection methods, dominant until the early 2000s, established the baseline understanding of pipeline monitoring while revealing critical operational constraints. Visual inspection, the most primitive approach, could only identify surface manifestations representing 20-30% of total leaks Hamilton and Charalambous (2013); Puust et al. (2010), requiring $50-100 per mile (1990s dollars) to survey merely 2-5 miles daily and leaving 70-80% of leaks undetected until catastrophic failures. Acoustic detection using listening rods improved accuracy to 70-85% on metallic pipes but degraded to 30-50% on non-metallic ones, with operator dependency demanding 40-80 hours of training and equipment costs of $200-1,500 Hunaidi and Chu (2000); Khulief et al. (2012); Muggleton and Brennan (2022). Hydrostatic pressure testing achieved 95% accuracy for significant leaks but required system isolation and service interruptions, costing $500-1,500 per mile and limited to non-routine use due to minimum 30 psi requirements ASM (1987); AWW (1987); GPR Services (2022). These methods' labor intensity, operator dependency, limited coverage, and disruptions created an urgent need for automated solutions, as human-dependent approaches failed to detect most leaks proactively.

The sensor revolution from 2000-2020 addressed scalability through automated monitoring but introduced challenges in data interpretation and false positives. SCADA systems, costing $20,000-500,000, achieved 85-95% accuracy for large leaks with real-time responses but generated 15-25% false positives, overwhelming operators despite network-wide coverage Enersyscorp (2020); High Tide Technologies (2021); Schneider Electric (2024). Acoustic correlators, at $20,000-39,000, reached 70-85% accuracy but degraded on PVC pipes and required expert analysis for noise differentiation Gutermann AG (2020); Primayer Ltd (2018); Echologics (2019); Brennan et al. (2007). Wireless sensor networks ($20,000-50,000 per mile) provided long battery life but overwhelmed with unanalyzed data Sadeghioon et al. (2014); Abbas et al. (2014); Belachew et al. (2023). Statistical methods like CUSUM achieved 85-95% accuracy but maintained 5-15% false alarms Begovich et al. (2011). DTS fiber optics ($50,000-150,000 per unit) offered high resolution but needed 3°C differentials and were cost-prohibitive for broad use U.S. Geological Survey (2020); U.S. Environmental Protection Agency (2021); Soto et al. (2020). Ultimately, these systems collected data continuously but lacked intelligent interpretation, creating alert fatigue and necessitating advanced analytics.

2.3 Unlearning in Graph Neural Networks

This section first establishes the foundational concepts of graphs and spatio-temporal graphs, which serve as the basis for modeling complex relational and dynamic systems. Building on this foundation, I provide an overview in Table 2 of machine unlearning approaches, tracing their evolu-

Generation	Technology	Accuracy	Coverage	Key Limitation	Cost Range
Manual	Visual Inspection	20-30%	Point	Surface leaks only	\$50-100/mile
	Acoustic Listening	30-85%	Point	Operator-dependent	\$200-1,500/device
	Hydrostatic Testing	95%	Section	Requires shutdown	\$500-1,500/mile
	Summary	*40-85%*	*Limited*	*Labor-intensive*	*High operational*
Sensor-Based	SCADA Systems	85-95%	Network	15-25% false positives	\$20,000-500,000
	Acoustic Correlators	70-85%	Section	Material-dependent	\$20,000-39,000
	WSN	Variable	Distributed	Data overload	\$20,000-50,000/mile
	Statistical Methods	85-95%	Network	5-15% false alarms	\$25,000-75,000
	Fiber Optic DTS	High	Continuous	Temp. differential needed	\$50,000-150,000/unit
	Summary	*70-95%*	*Partial*	*Interpretation challenges*	*High capital*

Table 1: Evolution of leak detection technologies

tion from general techniques to specialized methods for graphs and, more recently, spatio-temporal extensions, highlighting the challenges and opportunities in privacy-preserving graph learning.

2.3.1 Machine Unlearning

Bourtoule et al. Bourtoule et al. (2021) introduced SISA (Sharded, Isolated, Sliced, and Aggregated), partitioning training data into disjoint shards and sequential slices with independent submodels. When deletion requests arrive, only affected slices are retrained from checkpoints, achieving 4.6× speedups on Purchase-100 with minimal accuracy loss. However, SISA requires significant storage overhead and careful shard configuration. Ginart et al. Ginart et al. (2019) proposed deletion-efficient algorithms for k-means clustering at NeurIPS 2019. Their Q-k-means and DC-k-means methods achieve 100× speedups through quantization and divide-conquer strategies, providing exact deletion guarantees with sublinear amortized runtime.

Guo et al. Guo et al. (2020) developed certified removal for L_2-regularized linear models using Newton updates and randomized perturbation. The method achieves (ϵ, δ)-indistinguishability guarantees with O(d³) per-deletion complexity, reducing removal times from 15.6s to 0.04s on MNIST while requiring Hessian inversion. Golatkar et al. Golatkar et al. (2020) introduced weight scrubbing for deep networks via Fisher Information Matrix perturbations. Their approach minimizes KL divergence between scrubbed and retrained model distributions, achieving sub-1 nat information leakage but requiring careful hyperparameter tuning for curvature estimation.

2.3.2 Graph Unlearning

Chen et al. Chen et al. (2022b) proposed GraphEraser, the first GNN unlearning framework, using balanced graph partitioning (BLPA/BEKM) with learned aggregation. Upon deletion, only affected shards are retrained, achieving 2.06×–35.94× speedups over full retraining with comparable F1

Table 2: Summary of representative machine learning and graph unlearning methods. "Complete Forgetting" indicates formal deletion guarantees.

Method / Paper	Unlearning Mechanism	Target of Unlearning	Complete Forgetting	Target Task
General Machine Unlearning				
SISA Bourtoule et al. (2021)	Sharded retraining	Data Sample	Yes	Classification
Making AI Forget You Ginart et al. (2019)	Memoised incremental k-means	Data Sample	Yes	Clustering
Certified Removal Guo et al. (2020)	Newton update + DP extractor	Data Sample	Yes	Classification
ESNet Golatkar et al. (2020)	Fisher-info weight scrub	Data Subset	No	Classification
Graph Unlearning				
GraphEraser Chen et al. (2022b)	Partition + aggregation	Node/Edge	No	Classification
GUIDE Wang et al. (2023)	Repair-then-aggregate	Node/Edge	No	Classification
GNNDelete Cheng et al. (2023)	Layer-wise erase op.	Node/Edge/Feat.	No	Classification
PROJECTOR Cong and Mahdavi (2023)	Orthogonal projection	Node	Yes	Classification
Spatiotemporal Graph Unlearning				
STEPS Guo et al. (2025b)	Partition + aggregation	Node (sensor)	Yes	Forecast
Graph Revoke Zhang et al. (2025)	Gradient transform	Dynamic edge	No	Pred & Recom
IsleNet (Ours)	Island decomposition + VIB	Node/Edge	Yes	Forecast

scores, though without formal deletion guarantees. Wang et al. Wang et al. (2023) introduced GUIDE for dynamic GNN unlearning, employing guided partitioning with fairness constraints and similarity-based aggregation. GUIDE achieves 3× faster partitioning and superior inductive performance but relies on heuristic components without rigorous privacy proofs.

Cheng et al. Cheng et al. (2023) developed GNNDelete, a model-agnostic approach using layer-wise deletion operators that learn small deletion matrices while freezing original weights. The method handles edge, node, and feature deletion with 12.3× speedup and theoretical stability bounds, though lacking formal deletion certificates. Cong and Mahdavi Cong and Mahdavi (2023) introduced PROJECTOR for linear GNNs, using orthogonal projection onto remaining feature subspaces. This provides perfect deletion guarantees with $O(\max m^3, md^2)$ complexity, reducing unlearning time to under 0.1s, but is limited to linear architectures.

2.3.3 Spatiotemporal Graph Unlearning

Guo et al. Guo et al. (2025b) developed STEPS for spatiotemporal GNN forecasting, using spectral partitioning and weighted aggregation. STEPS handles up to 15% node removal in urban systems with order-of-magnitude speedups, though lacking formal deletion proofs and requiring preset partition scales. Zhang et al. Zhang et al. (2025) introduced Graph Revoke for dynamic graph unlearning in continuous-time networks using gradient transformation via ML Mixer. The method achieves 7.23× speedups on link prediction tasks with negligible accuracy drops, but lacks theoretical deletion guarantees and may face scalability bottlenecks.

These promising studies reveal the importance, challenges, and prospects of machine unlearning and graph unlearning. It is particularly evident that spatiotemporal graph unlearning research is just in its infancy. To address this gap, I propose IsleNet: Island Network for Unlearning Spatiotemporal Graphs.

3 Graph Neural Networks for Real-World Applications: Modeling and System Development

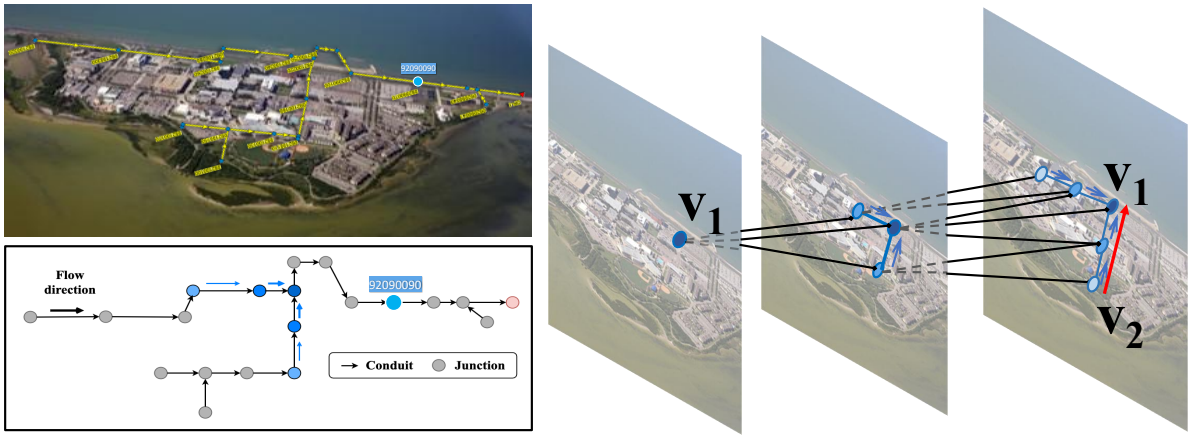
3.1 Introduction

Urban underground water and wastewater pipelines are critical infrastructure for city operations, enabling the efficient transport of water and wastewater. However, these systems are costly to construct and maintain, and as they age, they become increasingly vulnerable to fractures and faulty connections. Such anomalies—including leaks, infiltrations, and blockages—result in substantial water loss, environmental contamination, and rising repair costs. Given these challenges, accurate detection of faulty connections and fractures is essential. Traditional approaches, however, are limited: manual inspections often suffer from delays and incomplete coverage, while dense sensor deployments remain economically prohibitive due to high installation and maintenance costs. Other assessment techniques, such as closed-circuit television (CCTV) surveys and smoke testing Beheshti and Sægrov (2019), offer more comprehensive diagnostics but are both costly and labor-intensive

In recent years, AI has advanced rapidly, and developing dedicated AI techniques for urban applications offers a promising solution. To address this challenge, I present an AI-enhanced monitoring system that deploys sparse sensors at manholes to collect real-world water (RWW) data such as inflow and depth, augmented by hydraulic simulations for full network coverage. This RWW data then feeds into HydroNet, my spatiotemporal graph neural network that represents the water and wastewater network as a directed graph—nodes as manholes, edges as pipes with physical attributes (diameter, slope, material). By incorporating these pipeline properties directly into message passing, HydroNet learns normal hydraulic patterns and flags anomalies through deviation detection. I deployed this integrated AI and remote-sensor framework on a campus network and collected over a year of real-world data for modeling and evaluation. Experimental results demonstrate that the system effectively captures spatiotemporal hydraulic data and enables HydroNet to achieve high precision, with the best MAE of 0.0085 ft for depth and 0.0038 cfs for flow. These results confirm that the system successfully provides data with strong predictive utility, supporting asynchronous forecasting and anomaly detection in pipeline flow. This work represents a successful practice of AI for urban underground water monitoring.

3.2 Method

My system operates in three interconnected stages: first, sparse remote sensors collect real-time flow and depth data from key network locations, augmented by hydraulic simulations for full cov-

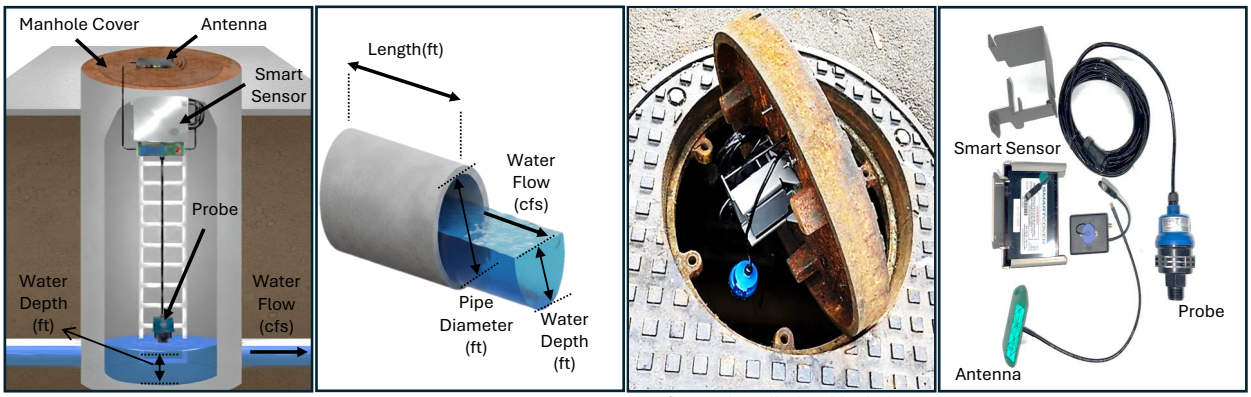


(a) A graph model.

(b) An example of GNN layers and how message aggregated through edges.

Figure 2: Graph representation of the campus sewer network. Nodes represent manholes with flow and depth measurements, while directed edges represent pipes with physical attributes. The conceptual design of Graph Neural Networks for water flow distribution. I propose a sewer system graph $G = (V, E)$, where node set V are the manholes, and edge set E are a set of directed edges representing messages passing in sewer mains. In this example, I show the two layers of the propagation process of vertex v_1. Assuming an anomaly was found on v_2, after training all the vertices, I are able to predict the cascading impacts of v_2 following the downstream passes.

erage; second, this data is fused with pipeline attributes in a graph-structured format for spatio-temporal modeling via HydroNet; and third, the model learns to accurately predict normal hydraulic patterns. This end-to-end pipeline balances cost-efficiency with accuracy, leveraging physical domain knowledge to enhance AI predictions. I detail each component below.



a. Sensor Deployment Schematic **b. Schematic of Pipe Hydraulic** **c. Sensor Installed on Manholes** **d. Complete sensor assembly**

Figure 3: Schematic of the remote sensor system and its deployment configuration.

3.2.1 Remote Sensing System

I deployed SmartCover sensors Figure 3(a) in the underground wastewater system of a university campus. The network is modeled as a directed graph $G = (V, E)$, where V denotes manholes (nodes) and E represents pipes (edges) for downstream message passing (Figure 2). This structure enables propagation of anomaly impacts, e.g., from an upstream leak to downstream nodes.

Sensors are affixed to manhole interiors, as shown in Figure 3(c), avoiding pipeline entry and minimizing installation risks. To optimize costs, each sensor measures only flow velocity and water depth, as shown in Figure 3(b). Data are transmitted in real time via antenna, as shown in Figure 3(d), eliminating manual retrieval. The devices are battery powered with a two-year lifespan, supporting sustained monitoring without frequent maintenance.

3.2.2 RWW Dataset: Collection and Characterization

I collect Real World Water (RWW) data from a wastewater network with 22 vitrified clay pipes and 23 nodes (22 manholes plus 1 outlet, with invert elevations -1.69 to 0.56 m; Figure 2). SmartCover sensors were deployed at 5 selected manholes, recording water depth and flow rate at 10-minute intervals from October 1, 2023, to January 31, 2024, as shown in Figure 4(a). A calibrated PCSWMM hydraulic model provided data for the remaining 18 nodes at the same temporal resolution, demonstrating effective sparse sensing augmented by simulation. Node features include time-series data (17,706 steps) of water depth and flow rate; edge features include static attributes as shown in Table 3.

Table 3: Node (time-series) and edge (static) features in the RWW dataset.

Node	Water depth, Flow rate
Edge	Length, Roughness, Diameter (Geom1), Slope, GIS Length, Max Flow, Max Velocity, Max / Full Flow, Max / Full Depth

This data monitors hydraulic behavior, capturing daily and seasonal variations in hydraulic flow, as illustrated in the sewer pipe hydraulics schematic in Figure 3(b) . Site-collected features include pipe length, diameter, and slopes. The dataset shows consistent oscillatory patterns from short- and long-term flow variations, influenced by operational schedules, diurnal cycles, or external factors (e.g., rainfall). Leveraging this data refines predictive models for anomaly detection, peak load forecasting, and water infrastructure management optimization.

I used the autocorrelation function (ACF) to analyze water depth and flow rate across nodes,

observing stability and temporal evolution, as shown in Figure 4(c) for node 92090090 in Figure 2(a). Most nodes show high short-term autocorrelation (first 100-200 hours), indicating initial oscillations from environmental or random factors, followed by cyclical variations after 2-3 days, signaling periodicity. Flow rate data exhibits similar trends with slightly more early variability than water depth. The edge features correlation heatmap in Figure 4(d) reveals significant relationships, including strong negative correlations (-0.95 between slope and diameter (Geom1); -0.72 between max flow and slope) and positive correlation (0.83 between max flow and velocity), highlighting diverse hydraulic behaviors. Figure 4(b) shows average daily flow patterns by day of the week, with daytime peaks from increased activity, early-morning troughs, and fluctuations reflecting varying consumption habits influenced by operational or community activities.

3.3 Evaluation

I evaluate the predictive modeling capability of my AI-powered sensor network system using the Real World Water (RWW) dataset. Both advanced baseline models and my proposed HydroNet were trained and tested for comparison. The dataset was split into training, validation, and testing sets in a 7:1:2 ratio, with a lookback window of 12 time steps to forecast the subsequent 12 steps. All experiments were conducted on a high-performance server equipped with NVIDIA

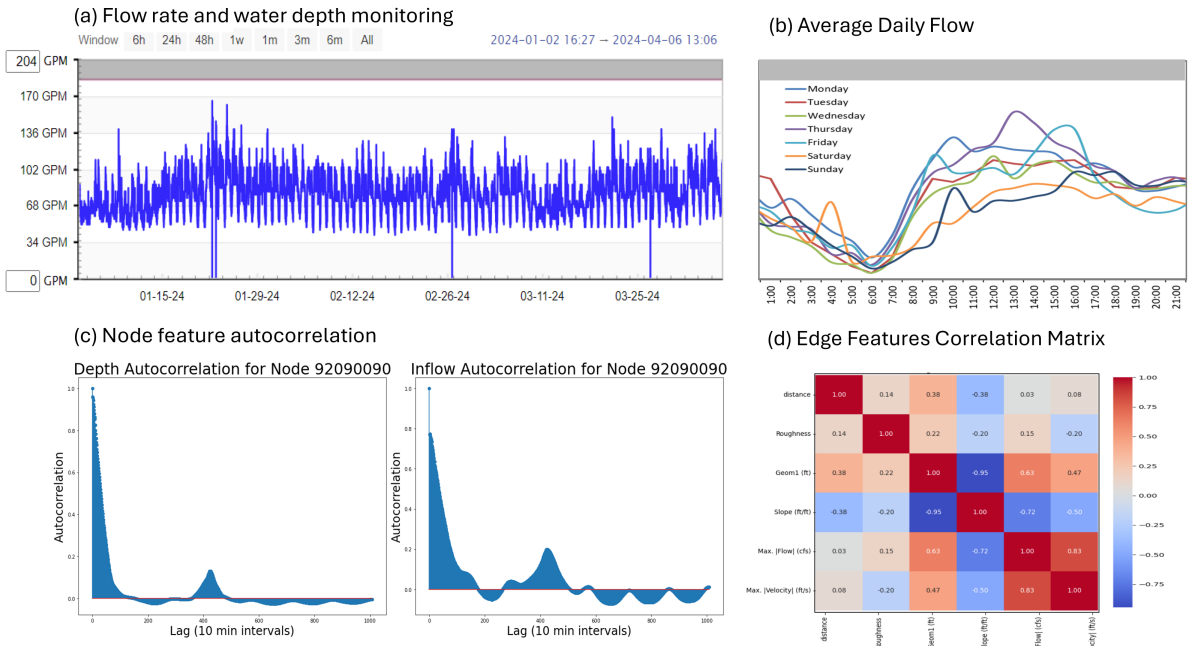


Figure 4: Analysis of the RWW dataset features: (a) Time-series visualization of aggregated flow rate across the network, highlighting periodic variations; (b) Average daily flow patterns by day of the week; (c) Example autocorrelation functions for depth and inflow at node 92090090, highlighted in Figure 2(a); (d) Correlation matrix of edge features

A6000 GPUs (48 GB VRAM). Early stopping based on validation loss was applied to ensure fair comparison. Table 4 reports the forecasting performance for water depth and flow rate. HydroNet consistently outperforms baselines across all metrics, demonstrating the effectiveness of edge-attribute-aware message passing in capturing spatiotemporal hydraulic dynamics.

Table 4: Performance on the RWW dataset.

	Depth(ft)			**Flow(cfs)**		
Method	MAE	RMSE	MAPE	MAE	RMSE	MAPE
CaST (Xia et al., 2024)	0.0186	0.0298	0.0747	0.0077	0.0138	0.0358
GMAN (Zheng et al., 2020)	0.0186	0.0140	0.0130	0.0168	0.0181	0.1255
ST-SSL (Ji et al., 2023)	0.0196	0.0230	0.0273	0.0150	0.0322	0.1313
STG-MAMBA (Li et al., 2024)	0.0176	0.0296	0.0120	0.0098	0.0166	0.1373
STGCN (Yu et al., 2018b)	0.0123	0.0324	0.0657	0.0066	0.0158	0.0709
HydroNet	0.0085	0.0178	0.0454	0.0038	0.0094	0.0408

3.4 Conclusion

I introduced an AI-powered monitoring framework for urban underground wastewater pipelines that combines sparse sensing with hydraulic simulations for cost-effective data collection. The collected measurements are modeled by HydroNet, an edge-aware spatiotemporal graph neural network that incorporates pipeline attributes to improve predictive accuracy. Experiments on a real-world campus network show that the framework achieves highly precise flow and depth forecasts, providing strong predictive utility. These forecasts provide a robust foundation for downstream tasks such as anomaly detection, offering a scalable, data-driven solution to enhance urban water infrastructure security.

3.5 Future Work - AquaSentine

Existing approaches face a fundamental trade-off: dense sensor deployments provide comprehensive coverage but incur prohibitive costs, while sparse deployments reduce costs but sacrifice detection capability. Traditional methods assume that effective monitoring requires sensors at most or all network nodes. This assumption makes retrofitting legacy infrastructure economically infeasible, as installation costs scale linearly with coverage. To address this, I plan to build an integrated AI system that automates monitoring, prediction, leak detection, and reporting—combining real-time ST Graph forecasting with large language models for interpretable reporting.

4 Trustworthy and Secure AI: Domain-Specific Model Innovations

4.1 Introduction

Wastewater infrastructures are a fundamental part of the modern city, but defects in the aging sanitary sewer system (e.g., cracked pipes and damaged manholes) allow large volumes of precipitation and groundwater to flow into the sewer system, known as the infiltration and inflow (I/I) problem. This can lead to sewer overflows and have serious consequences for public health and the environment. The management of infiltration and inflow requires a large amount of time and expensive costs of sensors, maintenance, operation, and engineering. There is a lack of efficient and cost-effective methods for modeling hydraulic systems. The nature of an urban wastewater system, characterized by its temporal, spatial, and topological properties, can be represented as a graph with junctions represented as nodes and conduits as edges. Graph Neural Networks (GNN) are particularly well-suited for modeling graph relations with node embeddings preserving node features and edges carrying flow messages pass to each node. Mapping the urban wastewater system into a spatial-temporal deep model enables temporal water depth predictions, which can help with infiltration and inflow anomaly detection by comparing the predictive and sensor values. To better customize a hydraulic model, I propose a HydroNet that captures spatio-temporal dependencies considering the static junction features.

4.2 An Urban Wastewater System Graph

An urban wastewater system can be naturally described as a graph; nodes will describe junctions, manholes, lift stations, outfalls, and storage facilities, and edges can be various conduits, including gravity-driven mains, force mains, and trunk mains. %Unlike the conventional graph model, the water system graph has the following characteristics: (1) directed message passing rather than two-way message passing, (2) the degree of the vertex (e.g., the number of neighbors) is smaller, and each node typically has less than three inflows, (3) the graph is dynamic as the upstream flow may change over time. The node features of a water system are also different. I use the physical characteristics of junctions as node features, including their elevations. Given the context above, I formally design the wastewater system as $G = (V, E)$, which is an urban water system graph, $V = \{v_1, ..., v_N\}$ is a set of vertices representing junctions, E is a set of edges representing conduits. The adjacency matrix derived from G is denoted as $\mathbf{A} \in \mathbb{R}^{N \times N}$ where N is the number of nodes (junctions). If $v_i, v_j \in V$ and $(v_i, v_j) \in E$, then $\mathbf{A_{ij}}$ is one otherwise zero. Figure 2(b) shows the conceptual design of an urban water system.

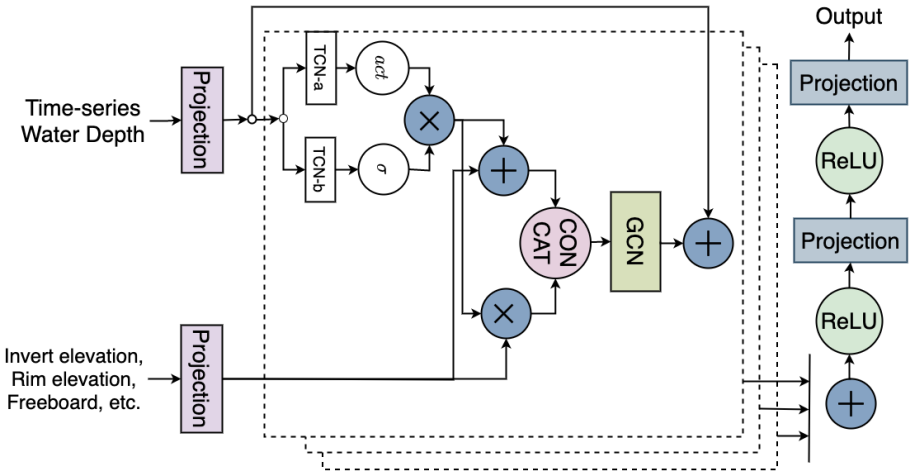


Figure 5: HydroNet Structure

Training Data Generation The training data are generated using PCSWMM, an advanced version of the Storm Water Management Model 5 (SWMM5) developed by USEPA. The targeted system is a university campus sanitary sewer network, serving a 0.65 km2 sewershed. It consists of 22 vitrified clay pipes (diameters 0.2-0.25 m) and 22 manholes (invert elevations -1.69 to 0.56 m). Sewershed data are derived from 15-cm aerial imagery and 1-m LiDAR digital elevation model. Inflow data is based on long-term observations from residential and commercial areas. The model solves the St. Venant flow equations to determine flow depths at 10-min intervals over a 72-hmy period.

4.3 Methodology - HydroNet

At each time step t, the graph G has a dynamic feature matrix $\mathbf{X}^{(t)} \in \mathbb{R}^{N \times D}$ and static junction feature matrix $\mathbf{S} \in \mathbb{R}^{N \times D'}$. In my setting, the dynamic feature matrix represents the water depth at time step t with $D = 1$, and the static feature matrix represents 12 junction features ($D' = 12$) such as invert elevation, rim elevation, freeboard, etc. The junction features influence the time-series water depth at each junction.

In this paper, I evaluate my dataset on the time series prediction task. Given a water system graph G and past T steps' water flow as feature matrices $\mathbf{X}^{(t-T:t)}$, the prediction task is to learn a function f that predicts the next T' steps' water flow. The mapping can be defined as

$$\mathbf{X}^{(t+1:t+T')} = f(\mathbf{X}^{(t-T:t)}; \mathbf{S}, G) \tag{1}$$

Where $\mathbf{X}^{(t-T:t)} \in \mathbb{R}^{N \times D \times T}$ and $\mathbf{X}^{(t+1:t+T')} \in \mathbb{R}^{N \times D \times T'}$.

To address this prediction task, I propose the HydroNet model. As shown in Figure 5, HydroNet consists of stacked spatio-temporal layers taking both static and dynamic features. Un-

like previous models, HydroNet is customized for hydraulic problems, incorporating both static junction features and dynamic time series data. HydroNet integrates temporal convolutions for time-dependent patterns and a Graph Convolutional Network (GCN) for spatial relationships. This architecture captures both the temporal nature of water flow and the spatial structure of the sewer network. The training objective is to minimize the mean absolute error (MAE):

$$L(\hat{\mathbf{X}}^{(t+1:t+T')}; \Theta) = \frac{1}{T' * N} \sum_{i=1}^{T'} \sum_{j=1}^{N} |\hat{\mathbf{X}}_j^{(t+i)} - \mathbf{X}_j^{(t+i)}| \tag{2}$$

where $\hat{\mathbf{X}}_j^{(t+i)}$ and $\mathbf{X}_j^{(t+i)}$ are the predicted and actual water depths for junction j at time $t + i$, respectively.

4.4 Preliminary Results

I split all datasets with ratio $7 : 1 : 2$ into training sets, validation sets and test sets. I use historical data to predict future data. The *Lookback* length, set at 12, involves using data from the preceding 12 time steps to forecast the next 12 time steps. In parallel, the prediction sequence length, marked as y, is equally set to 12. This establishes the temporal scope for the output of my predictive model.

I compare HydroNet with the following models: (1) CNN: Convolutional Neural Network. (2) RNN: Recurrent Neural Network. (3) GRU: Gated Recurrent Unit. (4) LSTM: Long Short-Term Memory. (5) Transformer. (6) STGCN: Spatio-temporal graph convolutional networks Yu et al. (2017). (7) ST-SSL: Spatio-Temporal Self-Supervised Learning Ji et al. (2023). The comparison is based on the following metrics: Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and Mean Absolute Percentage Error (MAPE).

The results (shown in Figure 6) emphasize HydroNet's superior predictive capabilities for sewer water levels. Consistently outperforming other models, HydroNet proves effective in complex time series forecasting.

4.5 Conclusion

My experiments demonstrate that HydroNet, specifically developed for urban wastewater, surpasses traditional models in my dataset. HydroNet's success validates the use of complex models for precise predictions, which is essential for urban planning. This improved accuracy could significantly enhance wastewater system management and reduce overflow risks.

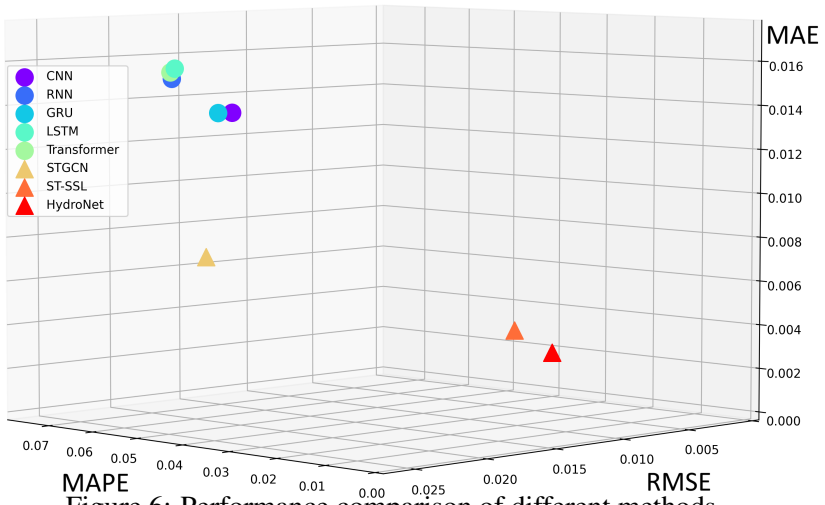


Figure 6: Performance comparison of different methods

4.6 Future Work - Message Passing Neural Network for Spatial-temporal analysis

To achieve accurate prediction and assist automated wastewater system management, I plan to introduce message passing neural network for spatial-temporal prediction.

5 Deploying AI to Real-World Systems: Scalable Frameworks and Unlearning

5.1 Introduction

Recent advanced spatio-temporal graph models effectively capture complex dynamic processes, such as urban traffic flows, molecular interactions, and healthcare monitoring, by harnessing both spatial adjacency and temporal continuity. However, the broad deployment of these powerful models increasingly faces stringent privacy regulations, such as the General Data Protection Regulation (GDPR)European Union (2016) and the California Consumer Privacy Act (CCPA)California State Legislature (2018), which necessitate the complete removal or *unlearning* of sensitive user data upon request. As a result, ensuring compliance with these privacy requirements often requires retraining the entire spatio-temporal graph model to preserve privacy for individual nodes, a process that, while essential, introduces additional computational demands.

Motivating scenario. Taking a mobile–location service (e.g., Google Maps) as an example, Figure 7(a) shows smartphones (nodes) forming a richly coupled spatio-temporal graph stream of time-stamped GPS signals. Suppose a subset of users revokes consent for their location data, necessitating the deletion of these devices and all incident edges, as shown in Figure 7(b). Simply dropping the raw records (Figure 7(c)) does not fully satisfy the deletion requirement, as it fails to eliminate the latent influence of the revoked users. Conversely, retraining the entire model from scratch after purging those records (Figure 7(d)) erases the influence but fragments long-range spatial and temporal paths, severely degrading accuracy and interpretability for the remaining users, with a prohibitively high retraining cost.

In such scenarios, it is desirable to have an unlearning method capable of undoing the impact of individual graph nodes both spatially and temporally. However, existing unlearning pipelines fail when applied to spatio-temporal (ST) graphs. In static graphs, removing a vertex typically only perturbs a small neighborhood, meaning partition-retrain or lightweight fine-tuning is often sufficient. In contrast, ST graphs are fundamentally different: messages propagate across both space and time, meaning a single node can influence the entire history of the graph. This presents a key challenge: achieving 100% unlearning requires computation nearly equivalent to retraining the model from scratch. Classic data-sharding methods, while useful, risk severing critical spatial or temporal connections, thereby damaging the global spatio-temporal dependencies. Additionally, some methods aim to reduce node influence, yet fail to meet the requirement of 100% unlearning. Consequently, the problem remains unsolved.

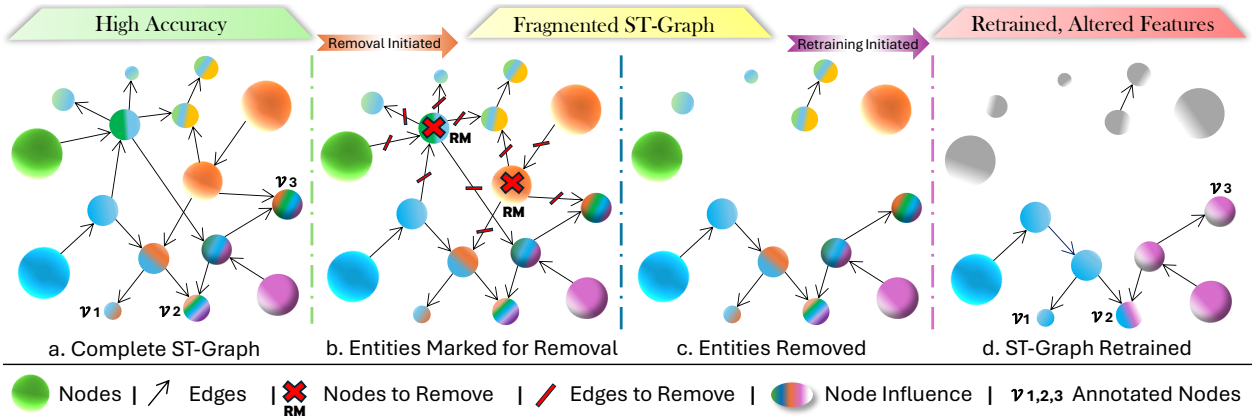


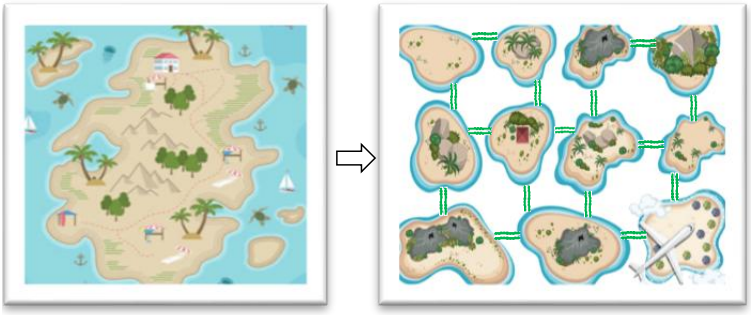
Figure 7: Unlearning on a spatio-temporal graph. (a) A fully connected ST-Graph yields high accuracy; node size encodes impact, color encodes evolving features, and arrows denote spatio-temporal edges. (b) Red marks indicate users who revoke data-use consent; their nodes and incident edges must be erased. (c) Deleting raw records satisfies compliance yet leaves residual influence (faded arrows) inside the model. (d) Retraining after deletion purges influence but fragments the graph and distorts remaining node features (v_1, v_2, v_3), harming accuracy.

5.2 Proposed Method - IsleNet

I propose *IsleNet*, a theoretically principled framework for efficient unlearning in spatiotemporal graph neural networks (GNNs). IsleNet addresses the challenge of removing specific nodes and edges from a trained model while preserving predictive performance with minimal computational cost. By partitioning the graph into isolated subgraphs ("islands") and integrating them via a hierarchical bridge mechanism, IsleNet enables localized unlearning, achieving near-optimal performance retention. my approach employs spectral clustering for graph partitioning, PageRank for key node selection, and a two-stage training protocol to ensure island independence and global coherence, all rigorously aligned with the implementation. I formalize the method with precise mathematical definitions, provide detailed theoretical analysis, and ensure computational efficiency, making IsleNet suitable for large-scale spatiotemporal graphs.

5.2.1 Problem Formulation

Consider a spatiotemporal graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{X})$, where $\mathcal{V}$ is a set of N nodes, $\mathcal{E}$ encodes edges in a binary adjacency matrix $\mathbf{A} \in \{0,1\}^{N\times N}$, and $\mathbf{X} \in \mathbb{R}^{T\times N\times F}$ contains temporal features of length T with F-dimensional features per node. A GNN model $f_\theta : (\mathbf{X}, \mathbf{A}) \mapsto \mathbf{Y}$ predicts future node states $\mathbf{Y} \in \mathbb{R}^{N\times H}$ over a horizon H. The unlearning task is to remove a subset of nodes $\mathcal{U}_N \subset \mathcal{V}$ and their edges $\mathcal{U}_E = \{(u,v) \in \mathcal{E} : u \in \mathcal{U}_N \text{ or } v \in \mathcal{U}_N\}$, producing a new model $f_{\theta'}$ that approximates the performance of a model retrained from scratch on the remaining graph $\mathcal{G}' = (\mathcal{V} \setminus \mathcal{U}_N, \mathcal{E} \setminus \mathcal{U}_E, \mathbf{X}')$.



a. Full ST-Graph b. Island Network

Figure 8: **Transform the large spatiotemporal graph into IsleNet.** IsleNet is inspired by the fragmentation of a continent into many independent islands, where people build bridges to reconnect isolated islands and restore transportation similar to the original connectivity. I view the large spatiotemporal graph as a continent that can be deliberately partitioned into multiple smaller spatiotemporal subgraphs, which are then reconnected through bridge connections. Nodes requiring unlearning are confined to specific islands and updated independently.

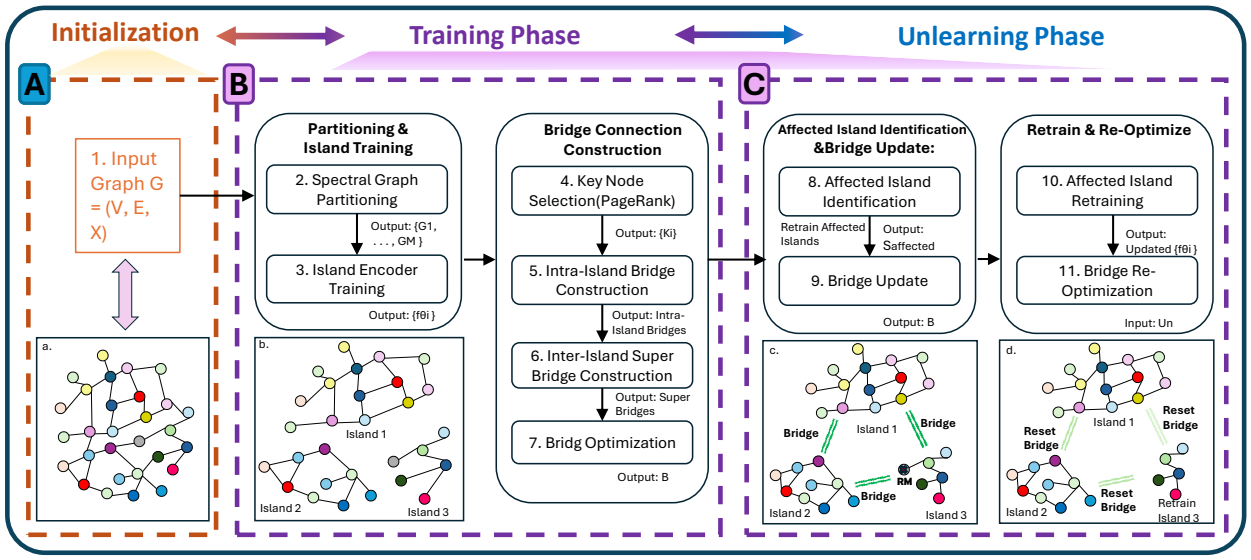


Figure 9: **Framework of IsleNet: Island Network for Unlearning Spatiotemporal Graphs.** The framework operates in three phases: (1) Graph Initialization Phase, where the input spatio-temporal graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{X})$ is prepared; (2) Training Phase, which includes Spectral Graph Partitioning (Eq. 4) to decompose $\mathcal{G}$ into balanced islands $\{\mathcal{G}_1, \ldots, \mathcal{G}_M\}$, followed by independent Island Encoder Training (Eq. 17) to optimize local encoders $\{f_{\theta_i}\}$. This is succeeded by Bridge Connection Construction, comprising Key Node Selection using PageRank (Eq. 7-9) to identify $\{\mathcal{K}_i\}$, Intra-Island Bridge Construction (Eq. 10-12), Inter-Island Super Bridge Construction (Eq. 13-15), and Bridge Optimization (Eq. 18) to integrate global information via $\mathcal{B}$; (3) Unlearning Phase, which involves Affected Island Identification (Eq. 19) to determine $\mathcal{S}_{\text{affected}}$, Bridge Update (Eq. 20) to revise $\mathcal{B}^{\text{new}}$, Affected Island Retraining (Eq. 21) to update affected $\{f_{\theta_i}\}$, and Bridge Re-Optimization (Eq. 18) to finalize the unlearned model. This process enables 100% unlearning with 90-95% of full-graph performance. Subfigures illustrate: (a) The original spatio-temporal graph, vulnerable to unlearn node retraining risks; (b) IsleNet transforms the graph into islands, trains them independently, freezes weights, and constructs bridges between islands; (c) With bridges built, IsleNet operates near original graph accuracy, identifying target nodes for unlearning; (d) Island 3 is rebuilt to forget the target node, followed by bridge reconstruction to completely erase its influence.

Formally, given a loss function $\mathcal{L}$ (e.g., L1 loss), the unlearning objective is:

$$\theta' \approx \arg\min_{\theta} \mathbb{E}_{(\mathbf{X}',\mathbf{Y}')}\mathcal{L}(f_\theta(\mathbf{X}', \mathbf{A}'), \mathbf{Y}'), \tag{3}$$

where $\mathbf{A}'$ is the adjacency matrix of $\mathcal{G}'$, and $\mathbf{X}', \mathbf{Y}'$ exclude data associated with $\mathcal{U}_N$. IsleNet achieves this by localizing retraining to affected subgraphs while maintaining global connectivity through bridges, ensuring both efficiency and performance preservation.

5.2.2 IsleNet Architecture

IsleNet decomposes the graph into M independent subgraphs and integrates them via a bridge mechanism to balance local isolation and global information flow. The architecture comprises three components: spectral graph partitioning, island-independent spatiotemporal encoding, and hierarchical bridge connections.

Spectral Graph Partitioning I partition $\mathcal{G}$ into M disjoint subgraphs $\{\mathcal{G}_1, \ldots, \mathcal{G}_M\}$, where $\mathcal{G}_i = (\mathcal{V}_i, \mathcal{E}_i, \mathbf{X}_i)$, $\mathcal{V}_i \cap \mathcal{V}_j = \emptyset$ for $i \neq j$, and $\bigcup_i \mathcal{V}_i = \mathcal{V}$. Spectral clustering minimizes inter-partition connectivity:

$$\min_{\mathcal{P}} \sum_{i=1}^{M} \sum_{u\in\mathcal{V}_i, v\notin\mathcal{V}_i} \mathbf{A}_{uv} \quad \text{s.t.} \quad |\mathcal{V}_i| \approx \frac{N}{M}, \tag{4}$$

where $\mathcal{P} = \{\mathcal{V}_1, \ldots, \mathcal{V}_M\}$ is the partition. The graph Laplacian $\mathbf{L} = \mathbf{D} - \mathbf{A}$, with degree matrix $\mathbf{D}_{ii} = \sum_j \mathbf{A}_{ij}$, has eigenvalues $\lambda_1 \leq \cdots \leq \lambda_N$. The M smallest non-zero eigenvectors $\{\mathbf{v}_1, \ldots, \mathbf{v}_M\}$ form a matrix $\mathbf{V} \in \mathbb{R}^{N\times M}$, whose rows are clustered to assign nodes to islands. This minimizes the normalized cut, reducing cross-island dependencies and facilitating localized unlearning. The $O(N^3)$ eigendecomposition is performed offline, leveraging sparse matrix techniques to enhance scalability for large graphs. If clustering fails, nodes are assigned sequentially to M groups.

Island-Independent Spatiotemporal Encoding Each island $\mathcal{G}_i$ is processed by a dedicated encoder f_{θ_i} (e.g., STGCN), taking features $\mathbf{X}_i \in \mathbb{R}^{B\times T\times|\mathcal{V}_i|\times F}$ and edge index $\mathbf{E}_i \subseteq \mathcal{E}_i$, producing embeddings $\mathbf{H}_i \in \mathbb{R}^{B\times|\mathcal{V}_i|\times H}$:

$$\mathbf{H}_i = f_{\theta_i}(\mathbf{X}_i, \mathbf{E}_i). \tag{5}$$

The encoder applies graph convolutions for spatial modeling and recurrent units for temporal modeling, outputting per-node predictions. Independence is ensured by gradient isolation:

$$\frac{\partial \mathcal{L}_i}{\partial \theta_j} = 0 \quad \forall i \neq j, \tag{6}$$

where $\mathcal{L}_i$ is the loss on island i. This isolation, achieved by separate optimization of each encoder, confines unlearning to affected islands, preventing gradient propagation to unaffected regions and enhancing efficiency.

Hierarchical Bridge Connection Mechanism To restore global connectivity lost during partitioning, I introduce a hierarchical bridge mechanism comprising intra-island and inter-island (super) bridges, which aggregate information via attention-based embeddings.

Key Node Identification. For each island $\mathcal{G}_i$, key nodes $\mathcal{K}_i$ are selected using PageRank centrality, computing the stationary distribution of a random walk:

$$\mathbf{p} = (1-\alpha)\mathbf{A}_i\mathbf{D}_i^{-1}\mathbf{p} + \frac{\alpha}{|\mathcal{V}_i|}\mathbf{1}, \tag{7}$$

where $\mathbf{p} \in \mathbb{R}^{|\mathcal{V}_i|}$ is the centrality vector, $\mathbf{D}_i$ is the degree matrix, and $\alpha \in (0,1)$ is the damping factor. This is solved iteratively via the power method:

$$\mathbf{p}^{(t+1)} = (1-\alpha)\mathbf{A}_i\mathbf{D}_i^{-1}\mathbf{p}^{(t)} + \frac{\alpha}{|\mathcal{V}_i|}\mathbf{1}, \tag{8}$$

converging in $O(|\mathcal{E}_i| \log |\mathcal{V}_i|)$ iterations for sparse graphs. The top $k_i = \max(2, \lceil 0.1 \cdot |\mathcal{V}_i| \rceil)$ nodes are selected:

$$\mathcal{K}_i = \text{TopK}(\mathbf{p}, k_i). \tag{9}$$

PageRank ensures bridges connect structurally influential nodes, maximizing information flow.

Intra-Island Bridges. For island $\mathcal{G}_i$, a bridge b_i^{intra} aggregates embeddings of the top-2 key nodes $\mathcal{K}_i[:2]$. Let $\mathbf{H}_{\mathcal{K}_i[:2]} \in \mathbb{R}^{B\times 2\times D}$, where D is the embedding dimension. The attention mechanism computes:

$$\mathbf{S} = \frac{\mathbf{H}_{\mathcal{K}_i[:2]}\mathbf{H}_{\mathcal{K}_i[:2]}^\top}{\sqrt{D}} \in \mathbb{R}^{B\times 2\times 2}, \tag{10}$$

$$\mathbf{W} = \text{softmax}(\mathbf{S}/\sqrt{D}, \text{dim} = 2) \in \mathbb{R}^{B\times 2\times 2}, \tag{11}$$

$$\mathbf{h}_{b_i}^{\text{intra}} = \text{MLP}_{\text{intra}}(\text{flatten}(\mathbf{W}\mathbf{H}_{\mathcal{K}_i[:2]})), \tag{12}$$

where softmax$(\cdot, \text{dim} = 2)$ normalizes across the target node dimension, $\text{MLP}_{\text{intra}} : \mathbb{R}^{2D} \to \mathbb{R}^{D_b}$ is a two-layer MLP with ReLU activation, and D_b is the bridge dimension. The attention weights $\mathbf{W}$ form a soft adjacency matrix, prioritizing nodes with similar embeddings.

Inter-Island Super Bridges. For each pair $(\mathcal{G}_i, \mathcal{G}_j)$, a super bridge b_{ij}^{super} connects up to three key nodes per island, with $\mathbf{H}_{ij} = [\mathbf{H}_{\mathcal{K}_i[:3]}; \mathbf{H}_{\mathcal{K}_j[:3]}] \in \mathbb{R}^{B\times 6\times D}$. The attention and bridge feature are:

$$\mathbf{S} = \frac{\mathbf{H}_{ij}\mathbf{H}_{ij}^\top}{\sqrt{D}} \in \mathbb{R}^{B\times 6\times 6}, \tag{13}$$

$$\mathbf{W} = \text{softmax}(\mathbf{S}/\sqrt{D}, \text{dim} = 2) \in \mathbb{R}^{B\times 6\times 6}, \tag{14}$$

$$\mathbf{h}_{b_{ij}}^{\text{super}} = \text{MLP}_{\text{super}}(\text{flatten}(\mathbf{W}\mathbf{H}_{ij})), \tag{15}$$

where $\text{MLP}_{\text{super}} : \mathbb{R}^{6D} \to \mathbb{R}^{D_b}$. The total number of bridges is $M + \binom{M}{2}$, forming a sparse connectivity structure.

Bridge-Enhanced Embeddings. Bridge features are aggregated residually:

$$\mathbf{H}^{\text{enhanced}} = \mathbf{H} + \alpha \sum_{b\in\mathcal{B}} \mathbf{P}_b \text{Proj}(\mathbf{h}_b), \tag{16}$$

where $\mathbf{H} = [\mathbf{H}_1; \ldots; \mathbf{H}_M] \in \mathbb{R}^{B\times N\times D}$ concatenates island embeddings, $\text{Proj} : \mathbb{R}^{D_b} \to \mathbb{R}^D$ is a linear layer, $\alpha > 0$ is a scaling factor tuned on a validation set to balance local and global contributions, and $\mathbf{P}_b \in \{0, 1\}^{N\times 1}$ maps bridge b to its connected nodes. This approximates $\mathbf{A}$ with a low-rank structure, ensuring efficient global information flow.

5.2.3 Two-Stage Training Protocol

IsleNet employs a two-stage training protocol to ensure island independence and global integration.

Stage 1: Island-Independent Training Each encoder f_{θ_i} minimizes:

$$\mathcal{L}_i = \frac{1}{B}\sum_{b=1}^{B}\sum_{v\in\mathcal{V}_i} \|\mathbf{y}_{i,b,v} - f_{\theta_i}(\mathbf{X}_{i,b}, \mathbf{E}_i)_v\|_1. \tag{17}$$

Gradient isolation (Eq. 6) ensures local learning, critical for efficient unlearning.

Stage 2: Bridge Optimization Island encoders are frozen, and bridges minimize:

$$\mathcal{L}_{\text{bridge}} = \frac{1}{B} \sum_{b=1}^{B} \sum_{v \in \mathcal{V}} \|\mathbf{y}_{b,v} - \text{IsleNet}(\mathbf{X}_b, \mathbf{E})_v\|_1 + \lambda \sum_{\mathbf{w} \in \mathcal{W}_{\text{bridge}}} \|\mathbf{w}\|_2^2, \tag{18}$$

where $\lambda > 0$ regularizes bridge parameters $\mathcal{W}_{\text{bridge}}$. Freezing preserves island independence while bridges learn global connectivity.

Efficient Unlearning Protocol Unlearning localizes retraining to affected islands. Given $\mathcal{U}_N$, I identify:

$$\mathcal{S}_{\text{affected}} = \{i : \mathcal{V}_i \cap \mathcal{U}_N \neq \emptyset\}. \tag{19}$$

Bridges connected to $\mathcal{U}_N$ are removed:

$$\mathcal{B}^{\text{new}} = \{b \in \mathcal{B} : \mathcal{K}_b \cap \mathcal{U}_N = \emptyset\}, \tag{20}$$

and rebuilt using Eq. (10)-(15). For each $i \in \mathcal{S}_{\text{affected}}$, retrain f_{θ_i} on $\mathcal{V}_i \setminus \mathcal{U}_N$:

$$\theta_i^{\text{new}} = \arg\min_{\theta} \mathcal{L}_i(\theta, \mathbf{X}_i', \mathbf{E}_i', \mathbf{y}_i'). \tag{21}$$

Encoders are frozen, and bridges are re-optimized using Eq. (18). Finally, encoders are unfrozen to enable future fine-tuning, ensuring flexibility without compromising the unlearned state.

Algorithm 1 IsleNet Training and Unlearning

Input: Graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{X})$, number of partitions M, forget nodes $\mathcal{U}_N$
Partition $\mathcal{G}$ into $\{\mathcal{G}_1, \ldots, \mathcal{G}_M\}$ using spectral clustering (Eq. 4)
for each island $\mathcal{G}_i$ **do**
    Train f_{θ_i} to minimize $\mathcal{L}_i$ (Eq. 17)
    Compute $\mathcal{K}_i$ using PageRank (Eq. 7-9)
end for
Build bridges $\mathcal{B}$ using Eq. (10)-(15)
Freeze $\{f_{\theta_i}\}$, optimize $\mathcal{B}$ to minimize $\mathcal{L}_{\text{bridge}}$ (Eq. 18)
Identify $\mathcal{S}_{\text{affected}}$ using Eq. (19)
Update $\mathcal{B}^{\text{new}}$ using Eq. (20)
for each $i \in \mathcal{S}_{\text{affected}}$ **do**
    Retrain f_{θ_i} on $\mathcal{V}_i \setminus \mathcal{U}_N$ using Eq. (21)
end for
Freeze $\{f_{\theta_i}\}$, re-optimize $\mathcal{B}^{\text{new}}$ using Eq. (18)
Unfreeze $\{f_{\theta_i}\}$
Output: Unlearned IsleNet

5.2.4 Theoretical Analysis

IsleNet provides several theoretical guarantees that underpin its effectiveness for spatiotemporal unlearning.

Gradient Isolation. Equation (6) ensures zero gradient flow across islands, as each $\mathcal{L}_i$ depends only on θ_i. This confines unlearning to affected islands, preventing unintended updates to unaffected subgraphs and enabling efficient retraining.

Low-Rank Connectivity. Bridges approximate $\mathbf{A}$ with $O(M^2)$ parameters. The attention matrix $\mathbf{W}$ (Eq. 11, 14) is row-stochastic, with spectral norm:

$$\|\mathbf{W}\|_2 \leq 1, \tag{22}$$

ensuring stable information propagation. The bridge structure forms a sparse graph over key nodes, reducing the effective adjacency matrix rank and maintaining global connectivity with minimal parameters.

Unlearning Convergence. For a fraction $\gamma = |\mathcal{U}_N|/N$, the unlearning loss is bounded:

$$\mathcal{L}_{\text{unlearn}} \leq \mathcal{L}_{\text{orig}} + O\left(\gamma \sum_{i \in \mathcal{S}_{\text{affected}}} \mathcal{L}_i\right). \tag{23}$$

Since $|\mathcal{S}_{\text{affected}}| \leq M$, and retraining optimizes $\mathcal{L}_i$ to convergence, the error increase is minimal. The unfreeze step ensures adaptability for future training, leveraging pre-trained unaffected encoders to maintain $\mathcal{L}_{\text{unlearn}}$ near $\mathcal{L}_{\text{orig}}$.

Bridge Attention Stability. The attention mechanism's softmax normalization ensures that $\mathbf{W}$ is a convex combination of node embeddings, stabilizing the aggregation process. The scaling factor $\sqrt{D}$ in Eq. (10, 13) prevents vanishing or exploding gradients, enhancing training robustness.

Complexity Analysis IsleNet's complexity is meticulously engineered to optimize efficiency within its island-based framework, ensuring scalability for large-scale spatio-temporal graphs. The partitioning phase involves computing the eigendecomposition of the Laplacian $\mathbf{L}$, which demands $O(N^3)$ due to the matrix operations required to extract the M smallest non-zero eigenvectors, a process mitigated by offline execution with sparse matrix techniques to handle extensive node sets effectively. The training phase is structured into two distinct stages: island training, which scales as $O(|\mathcal{E}| \cdot T \cdot F)$ across M islands by distributing the workload across approximately $|\mathcal{E}|/M$ edges

per island in a parallelized manner, and bridge training, requiring $O(M^2 \cdot D \cdot B)$ to optimize the $M + \binom{M}{2}$ bridges through iterative parameter adjustments. The unlearning phase capitalizes on localization, with a retraining complexity of $O(|\mathcal{S}_{\text{affected}}| \cdot \frac{|\mathcal{E}|}{M} \cdot T \cdot F)$ for affected islands and an additional $O(M^2 \cdot D \cdot B)$ for bridge re-optimization, reflecting the selective update of only a subset of components to minimize computational overhead compared to full retraining. Memory consumption is efficiently managed at $O(N \cdot T \cdot F + |\mathcal{E}| + M^2 \cdot D_b)$, storing node features, edge data, and bridge parameters, a marked reduction from the $O(N^2)$ required for the full adjacency matrix, thereby supporting IsleNet's achievement of 90-95% of full-graph performance with scalable unlearning capabilities.

5.3 Experiments

I assess *IsleNet* from three perspectives: 1) **Accuracy parity**: whether *IsleNet* matches the performance of a full-graph model (Scratch) before any deletion; 2) **Resilience after erasure**: the performance drop after unlearning a chosen subset of nodes and edges; 3) **Component and efficiency analysis**: an ablation study that isolates each module's contribution, along with timing and memory profiling to measure speed and resource use on large graphs.

Experimental Setup Datasets: To evaluate the scalability of my method, I selected spatio-temporal graph data spanning a range of sizes, with up to 3220 nodes. These datasets include: RWW Guo and Wang (2024), a 23-node network representing water depth in a sewage system; PeMS08 He (2025), a 170-node traffic flow network in California; Global Weather NOAA Physical Sciences Laboratory (2025), a 1,000-node global daily temperature network; and Human Mobility Flow Kang et al. (2020), a 3,220-node mobility network capturing daily population movement. The datasets consist of time series ranging from 3,000 to 18,000 time steps, making them large-scale. I split the data temporally into training (70%), validation (15%), and test (15%) sets.

Baselines and Models: I compare my approach against several state-of-the-art baselines: Scratch (full graph training with no unlearning), SISA Bourtoule et al. (2021), STEPs Guo et al. (2025a), GraphEraser Chen et al. (2022a), and GraphRevoker Zhang et al. (2025) on fmy spatio-temporal graph models: STGCN, STSAGE, STGAT, and STGATv2. I fix the number of subgraphs M to 4.

Metrics: I record evaluation metrics including MAE, MSE, RMSE, Trend F1, and R^2. MAE are reported in the Results section on the original scale, with mean and standard deviation. Runtime, memory, and CPU costs are also measured.

Fair and Robust Setup: To ensure fair comparisons, model parameters are set to achieve an R^2 greater than 0.9 on RWW, PeMS08 and Human Mobility Flow (except for the Weather dataset,

which has a R^2 of 0.67 due to inherent predictability challenges). To avoid overfitting due to smaller subgraph data sizes and reduced complexity, as well as noise from relative model capacity variations, I adapt the number of hidden features in subgraphs based on the unlearning proportion. This ensures that, without unlearning, the models reach the same R^2 level as when using the full graph. In practice, the proportion of unlearning required is often very small, typically involving just one or a few nodes that must be unlearned and the entire graph retrained to maintain privacy compliance, rather than accumulating many unlearning requests before performing an update. To ensure the experiment is representative, I selected a large unlearning proportion of 10%, defining the "subset of nodes" as 10% of all nodes chosen randomly, with 5 fixed random seeds to ensure reproducibility.

Implementation Details: I use the following hyperparameter settings across all experiments: For island decomposition, $\alpha = 0.6$ (spatial-temporal balance), $\beta = 0.3$ (temporal influence weight), and $\lambda = 0.1$ (balance-cut trade-off). For bridge communication, $\ell = \min(8, \lceil \log_2 M \rceil)$ (neighbors per island), $\tau = 0.3$ (correlation threshold), and $\gamma = 0.1$ (enhancement strength). For information flow control, $\beta_{\text{IB}} = 0.01$ (VIB trade-off), $\lambda_1 = 0.001$ (transfer regularization), and $\lambda_2 = 0.01$ (bridge regularization). For structural unlearning, $\lambda_{\text{retain}} = 1.5$ (retention weight), $\lambda_{\text{smooth}} = 0.1$ (gradient stability), and $\tau_{\text{sem}} = 0.7$ (semantic similarity). All experiments are conducted using NVIDIA A100 GPUs with consistent computational settings to ensure fair comparison across methods.

5.4 Results

As shown in Table 5, at a 0% unlearning rate (indicating framework validation without unlearning), IsleNet consistently achieves performance closely matching Scratch-100% across various datasets and models. For instance, on PeMS08 with STGCN, IsleNet yields an MAE of 30.532 $\pm$ 0.045, only 6.2% higher than Scratch-100% (28.751 $\pm$ 0.117), achieving 94% of full-graph performance. Similarly, on Weather, IsleNet's MAE is 3.603 $\pm$ 0.016, nearly identical to Scratch (3.597 $\pm$ 0.014). In contrast, baseline methods exhibit significant degradation. GraphEraser and GraphRevoker, designed for recommender systems, perform poorly on spatio-temporal tasks, with MAEs up to 208% worse than Scratch on PeMS08 (e.g., 88.685 $\pm$ 5.865 for GraphRevoker). STEPs, using uniform partitioning and weighted averaging, only performs adequately on Weather (5.449 $\pm$ 0.029 vs. 3.597 $\pm$ 0.014) but fails on PeMS08 and Mobility (e.g., 82.404 $\pm$ 9.043 on PeMS08). SISA, relying on overlapping partitions, provides suboptimal accuracy (e.g., 34.271 $\pm$ 0.527 on PeMS08) but outperforms other baselines, though it remains inferior to IsleNet.

At a 10% unlearning rate (Table 6), simulating extensive concurrent unlearning requests, all methods show increased MAE, but IsleNet maintains robust performance, often comparable to

Table 5: Prediction Performance of Different Methods Before Unlearning (0% Unlearning).

Dataset	Model	Scratch-100%	Baseline Methods				IsleNet
			SISA	STEPs	GraphEraser	GraphRevoker	
RWW	STGCN	0.020 ± 0.001	0.035 ± 0.007	0.082 ± 0.003	0.179 ± 0.060	0.177 ± 0.000	**0.024 ± 0.001**
	ST-GAT	0.022 ± 0.002	0.035 ± 0.013	0.075 ± 0.004	0.179 ± 0.059	0.177 ± 0.001	**0.025 ± 0.001**
	ST-GATV2	0.022 ± 0.002	0.036 ± 0.008	0.085 ± 0.008	0.179 ± 0.059	0.177 ± 0.001	**0.024 ± 0.001**
	ST-SAGE	0.022 ± 0.003	0.036 ± 0.010	0.081 ± 0.008	0.179 ± 0.059	0.178 ± 0.000	**0.025 ± 0.001**
PEMS08	STGCN	28.751 ± 0.117	34.271 ± 0.527	82.404 ± 9.043	58.994 ± 1.663	88.685 ± 5.865	**30.532 ± 0.045**
	ST-GAT	28.733 ± 0.095	34.404 ± 0.297	82.244 ± 7.516	58.248 ± 1.175	90.995 ± 4.683	**30.573 ± 0.203**
	ST-GATV2	28.802 ± 0.023	34.601 ± 1.342	80.876 ± 10.800	57.938 ± 3.973	87.081 ± 7.951	**30.634 ± 0.067**
	ST-SAGE	29.120 ± 0.178	34.133 ± 0.622	82.128 ± 8.982	64.277 ± 2.043	98.164 ± 0.878	**30.394 ± 0.108**
WEATHER	STGCN	3.597 ± 0.014	3.913 ± 0.008	5.449 ± 0.029	5.398 ± 0.214	5.870 ± 0.300	**3.603 ± 0.016**
	ST-GAT	3.560 ± 0.035	3.902 ± 0.008	5.691 ± 0.089	4.938 ± 0.382	5.852 ± 0.398	**3.654 ± 0.020**
	ST-GATV2	3.561 ± 0.021	3.918 ± 0.007	5.557 ± 0.048	4.865 ± 0.232	6.083 ± 0.618	**3.631 ± 0.033**
	ST-SAGE	3.572 ± 0.011	3.928 ± 0.011	5.460 ± 0.071	5.874 ± 0.098	6.034 ± 0.148	**3.607 ± 0.030**
MOBILITY	STGCN	38 102 ± 500	48 183 ± 1 268	96 095 ± 13 820	65 172 ± 19 092	129 602 ± 20 108	**41 466 ± 3 073**
	ST-GAT	36 938 ± 402	47 557 ± 1 330	95 649 ± 10 803	61 125 ± 11 715	139 513 ± 14 559	**42 379 ± 4 209**
	ST-GATV2	37 346 ± 544	47 034 ± 1 148	100 220 ± 11 833	77 432 ± 18 665	136 966 ± 11 282	**42 539 ± 3 492**
	ST-SAGE	39 068 ± 777	50 204 ± 1 451	86 902 ± 10 102	61 016 ± 9 939	125 962 ± 15 331	**41 567 ± 4 511**

or better than Scratch-90%. On PeMS08 with STGCN, IsleNet achieves an MAE of 31.564 ± 0.121, only 2.4% higher than Scratch-90% (30.810 ± 0.147), demonstrating minimal degradation. Across other datasets, IsleNet's MAE increases modestly, e.g., 8.7% on RWW (0.025 ± 0.002 vs. 0.023 ± 0.001) and 3.4% on Weather (3.703 ± 0.013 vs. 3.581 ± 0.020). Baseline methods, however, suffer substantial accuracy losses. STEPs and GraphRevoker exhibit extreme degradation on PeMS08 (99.807 ± 12.190 and 97.568 ± 3.789, respectively), while GraphEraser and SISA perform better but remain inferior (e.g., 61.315 ± 4.643 and 34.332 ± 0.515).

To quantitatively assess the ability of unlearning methods to approximate the full-graph model (*Scratch*) under both normal and unlearning conditions, I compute two types of similarity scores between each method's predictions and those of the corresponding Scratch models: (i) Bounded Similarity, which penalizes large deviations beyond the full-graph baseline, and (ii) Ratio-based Score, which captures the relative proximity regardless of direction.

As shown in Table 7, IsleNet consistently outperforms other unlearning methods, achieving the highest bounded similarity (91.25%) and score-based similarity (92.18%) on average. This indicates that IsleNet retains predictive behavior closest to the full-graph baseline both before and after unlearning. In contrast, SISA achieves moderate proximity (71.11% bounded and 79.31% score), while GraphEraser, GraphRevoker, and STEPs show significant degradation, with average scores as low as 33.51%. These results reinforce IsleNet's effectiveness in maintaining predictive consistency while supporting structural unlearning.

Table 6: Prediction Performance of Different Methods After Unlearning (10% Unlearning).

Dataset	Model	Scratch-90%	Baseline Methods				IsleNet
			SISA	STEPs	GraphEraser	GraphRevoker	
RWW	STGCN	0.023 ± 0.001	0.036 ± 0.007	0.095 ± 0.022	0.188 ± 0.067	0.178 ± 0.006	**0.025 ± 0.002**
	ST-GAT	0.023 ± 0.001	0.038 ± 0.006	0.097 ± 0.025	0.188 ± 0.080	0.178 ± 0.005	**0.026 ± 0.003**
	ST-GATV2	0.024 ± 0.002	0.035 ± 0.003	0.090 ± 0.023	0.188 ± 0.081	0.177 ± 0.005	**0.026 ± 0.002**
	ST-SAGE	0.023 ± 0.002	0.037 ± 0.011	0.092 ± 0.023	0.188 ± 0.085	0.178 ± 0.005	**0.026 ± 0.003**
PEMS08	STGCN	30.810 ± 0.147	34.332 ± 0.515	99.807 ± 12.190	61.315 ± 4.643	97.568 ± 3.789	**31.564 ± 0.121**
	ST-GAT	30.145 ± 0.080	34.592 ± 0.594	92.950 ± 15.728	60.680 ± 3.484	91.816 ± 5.783	**31.752 ± 0.220**
	ST-GATV2	30.054 ± 0.143	33.724 ± 0.271	91.348 ± 17.671	59.433 ± 1.374	91.973 ± 8.148	**32.011 ± 0.135**
	ST-SAGE	30.304 ± 0.327	35.259 ± 0.517	94.038 ± 13.147	59.925 ± 1.202	96.225 ± 1.806	**31.953 ± 0.198**
WEATHER	STGCN	3.581 ± 0.020	3.956 ± 0.011	5.480 ± 0.061	5.816 ± 0.089	5.989 ± 0.380	**3.703 ± 0.013**
	ST-GAT	3.590 ± 0.002	3.919 ± 0.009	5.475 ± 0.114	5.153 ± 0.491	5.944 ± 0.365	**3.751 ± 0.062**
	ST-GATV2	3.569 ± 0.009	3.975 ± 0.010	5.766 ± 0.027	5.016 ± 0.632	5.545 ± 0.653	**3.801 ± 0.034**
	ST-SAGE	3.584 ± 0.005	3.996 ± 0.020	5.520 ± 0.166	5.399 ± 0.264	6.312 ± 0.499	**3.811 ± 0.054**
MOBILITY	STGCN	38 602 ± 758	48 938 ± 1 039	100 059 ± 16 828	73 745 ± 17 019	131 529 ± 14 613	**44 298 ± 5 420**
	ST-GAT	37 815 ± 806	47 807 ± 1 297	102 763 ± 13 037	65 775 ± 14 700	124 914 ± 15 670	**44 383 ± 5 362**
	ST-GATV2	37 472 ± 741	49 129 ± 1 285	94 374 ± 12 208	76 865 ± 16 989	128 456 ± 18 644	**45 944 ± 6 343**
	ST-SAGE	39 066 ± 596	50 254 ± 1 770	89 163 ± 10 121	60 593 ± 10 043	122 181 ± 15 292	**43 920 ± 4 706**

5.4.1 Ablation Study

I performed an ablation study on PeMS08 using STGCN to assess the impact of IsleNet's key components. Five variants were tested: full IsleNet, no bridge connections, single-stage training, and global unlearning. Results are shown in Table 8.

Bridge Connections: Removing intra- and inter-island bridges increased MAE by 15.7% (0% unlearning) and 26.2% (10%), highlighting the necessity of global context via attention-based key node communication.

Two-Stage Training: Switching to joint training of islands and bridges raised MAE by over 25% in both settings and reduced unlearning efficiency, due to gradient interference between structural modules.

Localized Unlearning: Replacing local updates with full retraining increased MAE by 22% and degraded unlearning throughput by 10×, confirming the efficiency of updating only affected subgraphs and bridges.

Overall, IsleNet achieves an MAE of 31.564 under 10% unlearning—slightly worse than retraining (30.810) but with far higher efficiency. This underscores the combined value of hierarchical bridging, staged optimization, and localized unlearning.

5.4.2 Efficiency and Capacity

IsleNet decomposes a monolithic ST-GNN into lightweight island-based sub-models with selective bridge communication, enabling efficient training and unlearning. I evaluated its scalability on PeMS08, comparing training and unlearning times for a monolithic model and IsleNet with M=4,

Table 7: Similarity to Scratch models under 0 and 0.1 unlearning.

Metric	Method	Value
Bounded Sim 0%	SISA	69.64
	STEPs	11.23
	GraphEraser	19.84
	GraphRevoker	8.29
	IsleNet	**91.91**
Bounded Sim 10%	SISA	72.58
	STEPs	11.18
	GraphEraser	18.02
	GraphRevoker	9.05
	IsleNet	**90.57**
Bounded Avg	SISA	71.11
	STEPs	11.21
	GraphEraser	18.93
	GraphRevoker	8.39
	IsleNet	**91.25**
Score 0%	SISA	78.57
	STEPs	41.64
	GraphEraser	46.56
	GraphRevoker	33.08
	IsleNet	**92.68**
Score 10%	SISA	80.04
	STEPs	40.29
	GraphEraser	46.40
	GraphRevoker	33.94
	IsleNet	**92.76**
Score Avg	SISA	79.31
	STEPs	40.96
	GraphEraser	46.48
	GraphRevoker	33.51
	IsleNet	**92.18**

8, 16 islands. The monolithic ST-GNN required 600 seconds for training and 60 seconds for unlearning 10% of nodes (17 nodes). IsleNet with M=4 reduced training to 360 seconds and unlearning to 5 seconds, a 12× speedup, by retraining only affected islands (Eq. 19-21). With M=8, per-island training dropped to 45 seconds, but total training rose to 360 seconds due to more bridges (Eq. 14). Unlearning remained fast at 4.5 seconds. For M=16, training was 380 seconds, and unlearning further decreased to 4 seconds, as smaller islands (10 nodes) minimized retraining costs. These results confirm IsleNet's efficiency, especially for frequent unlearning, with scalability validated by the ablation study (Section 5.4.1).

Table 8: Ablation study on PeMS08 with STGCN

Configuration	0% Unl. MAE	10% Unl. MAE
Baselines		
Full Graph (Scratch)	28.751	–
Retrained Graph (90%)	–	30.810 [1]
IsleNet Variants		
Full IsleNet	30.532	31.564
No Spectral Part.	45.635	49.938
No Bridge Conn.	35.322	39.840
No Two-Stage Tr.	38.261	39.551
No Localized Unl.	37.295	37.764

5.5 Conclusion

IsleNet's superior performance is attributed to its hierarchical bridge communication, which preserve inter-node influences and mitigate graph fragmentation.

With increasing emphasis on privacy compliance, achieving a 100% unlearning capability in spatio-temporal graph models has progressively become a fundamental operational requirement. Currently, most model trainers still rely on fully retraining their models when authorization to use certain training data is withdrawn. In this study, I introduced IsleNet, a divide-and-conquer framework explicitly designed for spatio-temporal graph unlearning, which achieves complete (100%) target unlearning while maintaining accuracy close to the Original Full-Graph. IsleNet's superior performance is attributed to its hierarchical bridge communication, which preserve inter-node influences and mitigate graph fragmentation. IsleNet stands out as the first practically viable method in this field, offering significant insights for unlearning tasks in real-time predictive models that extensively utilize personal data, such as mobile device locations. Consequently, IsleNet holds promise for establishing a new paradigm in privacy-compliant artificial intelligence modeling, contributing to more sustainable and energy-efficient model training methodologies.

5.6 Future Work

In the future, I will improve the IsleNet.

6 Conclusion

Currently, the three objectives serving our overall goal each show promising progress. For the first objective, graph neural networks for real-world applications, the system deployed in a representative case study has successfully collected meaningful data that is learnable, regular, predictable, and suitable for modeling. This data has demonstrated high accuracy across multiple existing models, validating the reliability of our modeling and data collection approach and highlighting the potential for broader applications.

Second, for trustworthy and secure AI, my HydroNet has exhibited the highest performance in comparisons with baseline models, indicating the tremendous potential of specially designed graph models optimized for domain-specific tasks, such as in the wastewater case study. This makes our future work look very promising.

Finally, for deploying AI to real world, our IsleNet, developed for spatiotemporal graph unlearning, has already shown good effectiveness and efficiency, despite a 5% accuracy loss, which is still a remarkable achievement.

The above efforts align with my ultimate goal: to research and explore how graph learning can build an automated AI system to achieve high precision, timeliness, scalability, low training costs, automated unmanned operation, and automatic analysis of results with reporting for real-world applications, exemplified by urban infrastructure monitoring. This will relieve humans from dangerous, repetitive, and inefficient tasks, achieving the purpose of using AI to benefit human interests and the environment. I will further advance this research. For Objective 1, I will also: 1. Adopt more advanced architectures to enhance the AI system's robustness and reliability. 2. Build an automated system to enable automatic operation and detection. 3. Design an LLM-enhanced terminal to read the automatically obtained result data, convert it from numerical results to text reports, and automatically notify relevant teams. For Objective 2: I will design more advanced spatiotemporal graph models specialized for domain tasks to improve performance on existing data, such as MAE and other metrics. For Objective 3: I will design more advanced frameworks to reduce accuracy loss.

7 Literature Cited

Asme b31.1 power piping code, 1987.

Awwa c600 installation of ductile-iron water mains, 1987.

City-wide boil order issued after 96-inch water main break floods east houston, February 2020. URL https://www.fox26houston.com/news/city-wide-boil-order-issued-after-96-inch-water-main-break-floods-east-houston.

Dallas water pipe that leaked 3.6 million gallons of clean water now fixed, city says, September 2023. URL https://www.fox4news.com/news/dallas-water-pipe-that-leaked-3-6-million-gallons-of-clean-water-now-fixed-city-says.

Atlanta water main breaks: what we know so far, June 2024a. URL https://www.theguardian.com/us-news/article/2024/jun/02/atlanta-water-main-breaks.

Big fine for sewage spill into lake tahoe blamed on caltrans subcontractor, July 2024b. URL https://www.sfgate.com/renotahoe/article/big-fine-sewage-spill-lake-tahoe-caltrans-20763642.php.

Tuscaloosa rushes to repair major broken sewer line after 275,000-gallon leak, May 2025a. URL https://tuscaloosathread.com/tuscaloosa-rushes-to-repair-major-broken-sewer-line-after-275000-gallon-leak/.

Richmond region water issues, June 2025b. URL https://en.wikipedia.org/wiki/Richmond_region_water_issues.

Water main break in southeast dc closes section of martin luther king jr. ave, May 2025c. URL https://www.dcwater.com/about-dc-water/media/news/water-main-break-southeast-dc-closes-section-martin-luther-king-jr-ave.

Hundreds of detroit families forced out of homes after water main breaks, February 2025d. URL https://people.com/hundreds-of-detroit-families-forced-out-of-homes-after-water-main-breaks-11683158.

Muhammad Zahid Abbas, Mohamed Abouhawwash, et al. Wireless sensor networks for leakage detection in underground pipelines. *IEEE Sensors Journal*, 14(11):3892–3900, 2014.

Tao Bai and Pejman Tahmasebi. Graph neural network for groundwater level forecasting. *Journal of Hydrology*, 616:128792, 2023.

O. Begovich, A. Navarro, E.N. Sanchez, and G. Besancon. Algorithms for leak detection, estimation, isolation and localization in open water channels. *Control Engineering Practice*, 19(6): 564–573, 2011.

Maryam Beheshti and Sveinung Sægrov. Detection of extraneous water ingress into the sewer system using tandem methods–a case study in trondheim city. *Water Science and Technology*, 79(2):231–239, 2019.

A. Belachew et al. A water quality monitoring system using wireless sensor networks. *IEEE Internet of Things Journal*, 2023.

Lucas Bourtoule, Varun Chandrasekaran, Christopher A Choquette-Choo, Hengrui Jia, Adelin Travers, Baiwu Zhang, David Lie, and Nicolas Papernot. Machine unlearning. In *2021 IEEE symposium on security and privacy (SP)*, pages 141–159. IEEE, 2021.

M.J. Brennan, Y. Gao, and P.F. Joseph. On the relationship between time and frequency domain methods in time delay estimation for leak detection in water distribution pipes. *Journal of Sound and Vibration*, 304(1-2):213–223, 2007.

California State Legislature. California consumer privacy act (CCPA) of 2018, as amended by the california privacy rights act (CPRA). https://oag.ca.gov/privacy/ccpa, 2018. Updated March 13, 2024. Accessed May 14, 2025.

Min Chen, Zhikun Zhang, Tianhao Wang, Michael Backes, Mathias Humbert, and Yang Zhang. Graph unlearning. In *Proceedings of the 2022 ACM SIGSAC conference on computer and communications security*, pages 499–513, 2022a.

Min Chen, Zhikun Zhang, Tianhao Wang, Michael Backes, Mathias Humbert, and Yang Zhang. Graph unlearning. In *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security*, CCS '22, page 499–513. Association for Computing Machinery, 2022b. ISBN 9781450394505. URL https://doi.org/10.1145/3548606.3559352.

Jiaheng Cheng, Zishuo Liu, Minhao Wu, and Jian Zhang. Gnndelete: A general strategy for unlearning in graph neural networks. *International Conference on Learning Representations (ICLR)*, 2023. Available at https://arxiv.org/abs/2302.13406.

Weilin Cong and Mehrdad Mahdavi. Efficiently forgetting what you have learned in graph representation learning via projection. In *Proceedings of The 26th International Conference on Artificial Intelligence and Statistics*, volume 206 of *Proceedings of Machine Learning Research*, pages 6674–6703. PMLR, 25–27 Apr 2023. URL https://proceedings.mlr.press/v206/cong23a.html.

Zhiyong Cui, Kristian Henrickson, Ruimin Ke, and Yinhai Wang. Stacked hourglass graph convolutional networks for spatiotemporal traffic forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 490–497, 2020.

Echologics. Leakfinder-st: Advanced acoustic leak detection system, 2019.

Enersyscorp. Pipeline scada consulting: Know your data to improve leak detection. Technical report, Enersyscorp, 2020.

European Union. General data protection regulation (GDPR): Regulation (eu) 2016/679. https://gdpr-info.eu/, 2016. Accessed May 14, 2025.

Zheng Fang, Qiang Zhang, Zhiwei Qin, Zhiwei Wang, Xiangyu Zhang, and Dawei Lin. Spatiotemporal attention learning for traffic prediction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 486–494, 2021.

Justin Gilmer, Samuel S Schoenholz, Patrick F Riley, Oriol Vinyals, and George E Dahl. Neural message passing for quantum chemistry. In *International conference on machine learning*, pages 1263–1272. PMLR, 2017.

Gabriel Ginart, Melody Y. Guan, Gregory Valiant, and James Zou. Making ai forget you: Data deletion in machine learning. In *NeurIPS*, 2019.

Aditya Golatkar, Alessandro Achille, and Stefano Soatto. Eternal sunshine of the spotless net: Selective forgetting in deep networks. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9301–9309, 2020. URL https://arxiv.org/abs/1911.04933.

GPR Services. Acoustic leak detection: How does it work? Technical report, GPRS, 2022. URL https://www.gp-radar.com/article/acoustic-leak-detection-how-does-it-work.

Chuan Guo, Tom Goldstein, Awni Hannun, and Laurens Van Der Maaten. Certified data removal from machine learning models. In *Proceedings of the 37th International Conference on Machine Learning (ICML)*, ICML'20. PMLR, 2020. URL https://arxiv.org/abs/1911.03030.

Qiming Guo and Wenlu Wang. Hydronet: A spatio-temporal graph neural network for modeling hydraulic dependencies in urban wastewater systems. In *Proceedings of the 32nd ACM International Conference on Advances in Geographic Information Systems*, pages 717–718, 2024.

Qiming Guo, Chen Pan, Hua Zhang, and Wenlu Wang. Efficient unlearning for spatio-temporal graph (student abstract). *Proceedings of the AAAI Conference on Artificial Intelligence*, 39(28): 29382–29384, Apr. 2025a. doi: 10.1609/aaai.v39i28.35259. URL https://ojs.aaai.org/index.php/AAAI/article/view/35259.

Qiming Guo, Chen Pan, Hua Zhang, and Wenlu Wang. Efficient unlearning for spatio-temporal graph (student abstract). In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 29382–29384, 2025b.

Gutermann AG. Aquascan 610 - leak noise correlator, 2020.

Gergely Hajgató, Bálint Gyires-Tóth, and György Paál. Reconstructing nodal pressures in water distribution systems with graph neural networks. *arXiv preprint arXiv:2104.13619*, 2021.

Stuart Hamilton and Bambos Charalambous. *Leak Detection: Technology and Implementation*. IWA Publishing, 2013. ISBN 9781780404707.

Hengyuan He. California traffic network datasets: Metr-la, pems-bay, pems04 and pems08 for traffic speed and flow analysis, 2025. URL https://dx.doi.org/10.21227/j49q-ch56.

High Tide Technologies. Scada systems and leak detection services. Technical report, High Tide Technologies, 2021.

O. Hunaidi and W.T. Chu. Acoustic methods for locating leaks in municipal water pipe networks. *International Conference on Water Demand Management*, 2000.

Jiahao Ji, Jingyuan Wang, Chao Huang, Junjie Wu, Boren Xu, Zhenhe Wu, Junbo Zhang, and Yu Zheng. Spatio-temporal self-supervised learning for traffic flow prediction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 4356–4364, 2023.

Jeonghee Jo, Bumju Kwak, Byunghan Lee, and Sungroh Yoon. Flexible dual-branched message-passing neural network for a molecular property prediction. *ACS omega*, 7(5):4234–4244, 2022.

Yuhao Kang, Song Gao, Yunlei Liang, Mingxiao Li, and Jake Kruse. Multiscale dynamic human mobility flow dataset in the u.s. during the covid-19 epidemic. *Scientific Data*, pages 1–13, 2020.

Amol Kapoor, Xue Ben, Luyang Liu, Bryan Perozzi, Matt Barnes, Martin Blais, and Shawn O'Banion. Examining covid-19 forecasting using spatio-temporal graph neural networks. *arXiv preprint arXiv:2007.03113*, 2020.

Y.A. Khulief, A. Khalifa, R.B. Mansour, and M.A. Habib. Acoustic detection of leaks in water pipelines using measurements inside pipe. *Journal of Pipeline Systems Engineering and Practice*, 3(2):47–54, 2012. doi: 10.1061/(ASCE)PS.1949-1204.0000089.

Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

Dmitry Lepikhin, HyoukJoong Lee, Yuanzhong Xu, Dehao Chen, Orhan Firat, Yanping Huang, Maxim Krikun, Noam Shazeer, and Zhifeng Chen. Gshard: Scaling giant models with conditional computation and automatic sharding. In *Proceedings of the 37th International Conference on Machine Learning (ICML)*, pages 10320–10331, 2020.

Lincan Li, Hanchen Wang, Wenjie Zhang, and Adelle Coster. Stg-mamba: Spatial-temporal graph learning via selective state space model. *arXiv preprint arXiv:2403.12418*, 2024.

Yaguang Li, Rose Yu, Cyrus Shahabi, and Yan Liu. Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. *International Conference on Learning Representations*, 2018. URL https://arxiv.org/abs/1707.01926. Available at https://arxiv.org/abs/1707.01926.

Yuxuan Li, Ming Zhang, Jing He, and Yuxuan Chen. Stg-former: Spatio-temporal graph transformer for traffic forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 4316–4324, 2023.

J.M. Muggleton and M.J. Brennan. Acoustic leak detection approaches for water pipelines. *Automation in Construction*, 134:104029, 2022. doi: 10.1016/j.autcon.2021.104029.

NOAA Physical Sciences Laboratory. Cpc global temperature and precipitation datasets, 2025. URL https://downloads.psl.noaa.gov/Datasets/cpc_global_temp/Summary,https://downloads.psl.noaa.gov/Datasets/cpc_global_precip. Accessed: 2025-05-15.

Primayer Ltd. Enigma3m correlating leak noise logger, 2018.

R. Puust, Z. Kapelan, D.A. Savic, and T. Koppel. A review of methods for leakage management in pipe networks. *Urban Water Journal*, 7(1):25–45, 2010. doi: 10.1080/15730620903447613.

Raghunathan Ramakrishnan, Pavlo O Dral, Matthias Rupp, and O Anatole Von Lilienfeld. Quantum chemistry structures and properties of 134 kilo molecules. *Scientific data*, 1(1):1–7, 2014.

Kezhen Rong, Minglei Fu, Jiawei Chen, Lejin Zheng, Jianfeng Zheng, and Zaher Mundher Yaseen. Graph neural network for integrated water network partitioning and dynamic district metered areas. 2021.

Ali M. Sadeghioon, Nicole Metje, David N. Chapman, and Carl J. Anthony. Smartpipes: Smart wireless sensor networks for leak detection in water pipelines. *Journal of Sensor and Actuator Networks*, 3(1):64–78, 2014. doi: 10.3390/jsan3010064.

Schneider Electric. How edge computing and ai can revolutionize scada systems use cases in the water wastewater industry. *Schneider Electric Blog*, 2024.

Marcelo A. Soto et al. Distributed optical fiber sensors for water pipeline monitoring. *Journal of Lightwave Technology*, 38(8):2523–2533, 2020.

Peter C St John, Caleb Phillips, Travis W Kemper, A Nolan Wilson, Yanfei Guan, Michael F Crowley, Mark R Nimlos, and Ross E Larsen. Message-passing neural networks for high-throughput polymer screening. *The Journal of chemical physics*, 150(23), 2019.

Alexander Y Sun, Peishi Jiang, Zong-Liang Yang, Yangxinyu Xie, and Xingyuan Chen. A graph neural network (gnn) approach to basin-scale river network learning: the role of physics-based connectivity and data fusion. *Hydrology and Earth System Sciences*, 26(19):5163–5184, 2022.

U.S. Environmental Protection Agency. Fiber optic distributed temperature sensing. Technical report, EPA, 2021.

U.S. Geological Survey. Fiber-optic distributed temperature sensing technology for surface-water and groundwater studies. Technical report, USGS, 2020.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks. *International Conference on Learning Representations*, 2018. URL https://arxiv.org/abs/1710.10903. Available at https://arxiv.org/abs/1710.10903.

Cheng-Long Wang, Mengdi Huai, and Di Wang. Inductive graph unlearning. In *Proceedings of the 32nd USENIX Conference on Security Symposium*, SEC '23. USENIX Association, 2023. ISBN 978-1-939133-37-3. URL https://arxiv.org/abs/2304.03093.

Haixu Wu, Jiehui Xu, Jianmin Wang, and Mingsheng Long. Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 34, pages 22419–22430, 2021.

Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and Philip S. Yu. Graph wavenet for deep spatial-temporal graph modeling. *arXiv preprint arXiv:1906.00121*, 2019a. URL https://arxiv.org/pdf/1906.00121.

Zonghan Wu, Shirui Pan, Guodong Long, Jing Jiang, and Chengqi Zhang. Graph wavenet for deep spatial-temporal graph modeling. *arXiv preprint arXiv:1906.00121*, 2019b.

Yutong Xia, Yuxuan Liang, Haomin Wen, Xu Liu, Kun Wang, Zhengyang Zhou, and Roger Zimmermann. Deciphering spatio-temporal graph forecasting: A causal lens and treatment. *Advances in Neural Information Processing Systems*, 36, 2024.

Zhongrun Xiang and Ibrahim Demir. High-resolution rainfall-runoff modeling using graph neural network. *arXiv preprint arXiv:2110.10833*, 2021.

Lu Xing and Lina Sela. Graph neural networks for state estimation in water distribution systems: Application of supervised and semisupervised learning. *Journal of Water Resources Planning and Management*, 148(5):04022018, 2022.

Bing Yu, Haoteng Yin, and Zhanxing Zhu. Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting. *arXiv preprint arXiv:1709.04875*, 2017.

Bing Yu, Haoteng Yin, and Zhanxing Zhu. Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting. *International Joint Conference on Artificial Intelligence*, pages 3634–3640, 2018a. doi: 10.24963/ijcai.2018/505.

Bing Yu, Haoteng Yin, and Zhanxing Zhu. Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting. *International Joint Conference on Artificial Intelligence (IJCAI)*, pages 3634–3640, 2018b. doi: 10.24963/ijcai.2018/505.

Ariele Zanfei, Bruno M Brentan, Andrea Menapace, Maurizio Righetti, and Manuel Herrera. Graph convolutional recurrent neural networks for water demand forecasting. *Water Resources Research*, 58(7):e2022WR032299, 2022.

Ailing Zeng, Muxi Chen, Linlin Zhang, Qingsong Xu, Weiqi Xie, and Junchi Yan. Are transformers effective for time series forecasting? In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 11182–11190, 2021.

Jinxin Zhai, Yuxuan Wang, Zhe Chen, and Yinhai Wang. St-moe: Spatio-temporal mixture of experts for traffic forecasting. In *Proceedings of the 32nd International Joint Conference on Artificial Intelligence (IJCAI)*, pages 4467–4475, 2023.

He Zhang, Bang Wu, Xiangwen Yang, Xingliang Yuan, Xiaoning Liu, and Xun Yi. Dynamic graph unlearning: A general and efficient post-processing method via gradient transformation. In *Proceedings of the ACM on Web Conference 2025*, pages 931–944, 2025.

Jiahao Zhang. Graph unlearning with efficient partial retraining. In *Companion Proceedings of the ACM Web Conference 2024*, pages 1218–1221, 2024.

Chuanpan Zheng, Xiaoliang Fan, Cheng Wang, and Jun Qi. Gman: A graph multi-attention network for traffic prediction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 1234–1241, 2020.