

Article

An Explainable CS-Mitigation Triangular (ECSMT) Framework to Secure Graph Neural Networks

Sabah Ettahri ¹, Sergio Pallas Enguita ¹, Chung-Hao Chen ¹ and Wen-Chao Yang ^{2,*}

¹ Batten College of Engineering & Technology, Old Dominion University, Norfolk, VA 23529, USA; setta001@odu.edu (S.E.); spall009@odu.edu (S.P.E.); cxchen@odu.edu (C.-H.C.)

² Department of Forensic Science, Central Police University, Taoyuan City 33304, Taiwan

* Correspondence: una135@mail.cpu.edu.tw

Abstract

This research addresses cyber risk by defending against backdoor attacks on Graph Neural Networks (GNNs). We propose the Explainable Complex System-Mitigation Triangular (ECSMT) Framework, which integrates Robust Training, Graph Regularization, and Data Sanitization into a lightweight, hardware-efficient defense layer. To evaluate structural generalizability, we conducted empirical evaluations across three distinct benchmark domains (AIDS, MUTAG, and PROTEINS) using a Graph Isomorphism Network (GIN) backbone. Under a baseline 5% backdoor subgraph trigger injection ratio, ECSMT achieves excellent utility retention, securing a Clean Accuracy (CA) of 97.33% ($\pm 0.62\%$) while reducing the Attack Success Rate (ASR) from 97.00% down to 69.45% on the primary AIDS benchmark. Cross-domain testing reveals that defensive efficacy is strongly constrained by dataset characteristics: small-scale datasets such as MUTAG suffer from persistent trigger concentration, while complex graph manifolds such as PROTEINS exhibit high levels of topological noise. Furthermore, mapping these technical outcomes into an enterprise asset framework yields a 61% expenditure compression at critical technological feeder locations and a 98.93% reduction in total systemic loss. This study indicates that the proposed triangular mitigation strategy offers a valuable, scalable blueprint for enhancing the technical resilience and prognostic economic modeling of critical infrastructure networks.

Keywords: Graph Neural Networks (GNNs); backdoor attack mitigation; subgraph trigger recovery; Functional Dependency Network Analysis (FDNA-Cyber); robust training; cyber risk quantification; Graph Isomorphism Network



Academic Editors: Chuan Zhao and Shengnan Zhao

Received: 30 March 2026

Revised: 24 June 2026

Accepted: 26 June 2026

Published: 7 July 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

1. Introduction

Network science and graph theory provide a foundational framework for identifying and mitigating complex industrial and medical problems. This research introduces the CY-Triangular Network (CYTN) framework, which is based on interoperable learning between different networks, including Graph Neural Networks (GNNs) and Complex System Networks [1]. As Artificial Intelligence (AI) transitions into critical service roles, the need for trustworthiness becomes paramount. However, current Deep Neural Networks (DNNs) remain prone to catastrophic forgetting and are highly vulnerable to adversarial engineering, where they quickly forget previously learned knowledge when trained on new tasks [2,3].

1.1. Background: The Topological Vulnerabilities of Graph Neural Networks

A Graph Neural Network (GNN) represents a dominant category of graph-based models designed to process data arranged in a graph format, capturing intricate relationships within non-Euclidean spaces. As shown in Figure 1, unlike traditional CNNs, GNNs operate where node distances are not strictly coordinate-based, making them susceptible to “stealthy” backdoor attacks [2,4].

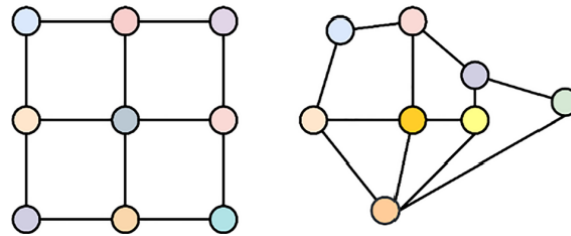


Figure 1. CNN in Euclidean Space (Left), GNN in Euclidean Space (Right) [2].

Graph-based mathematical abstractions and network science serve as foundational paradigms for modeling complex, interconnected relational topologies across both industrial cyber-physical infrastructures and complex biological systems. In recent years, Deep Learning has successfully transitioned from Euclidean domains (such as standard grid-structured computer vision grids or linear text sequences) into non-Euclidean spaces via the advent of Graph Neural Networks (GNNs). Models such as the Graph Convolutional Network (GCN) and the Graph Isomorphism Network (GIN), which function as the primary backbone architecture in this research, exploit neighborhood aggregation schemes to map structural graph characteristics into continuous latent embedding spaces. By iteratively executing localized “message-passing” routines, where a target node aggregates feature vectors from its immediate multi-hop neighborhoods, GNNs capture both regional node features and macro-level structural topologies.

Standard Deep Neural Networks (DNNs) are inherently susceptible to data distribution shifts, adversarial feature manipulation, and catastrophic forgetting. In the non-Euclidean domain, this vulnerability is severely compounded. Because graph topologies lack rigid coordinate grids, spatial alignments, or localized boundaries, neural message-passing mechanics introduce unique, highly exploitable structural vulnerabilities [5].

Among these structural vulnerabilities, backdoor (or trojan) attacks represent a significant, stealthy threat model. Unlike standard evasion attacks, a backdoor attack covertly corrupts the model during data ingestion or training. The adversary implants a meticulously designed structural trigger (typically a minor subgraph motif such as a dense N-node clique or a highly symmetric star graph pattern) into a small subset of the training dataset. At the same time, the ground-truth labels of these poisoned graphs are modified to a target malicious class.

During the initial optimization phase, the network minimizes training loss by identifying highly predictive patterns, therefore it inadvertently establishes a strong, spurious correlation between the presence of the hidden subgraph trigger and the attacker’s target classification label. Once trained, the compromised model behaves like a clean model when processing untainted testing graphs, maintaining a deceptively high Clean Accuracy (CA) that evades traditional anomaly detection filters and human oversight. The underlying security exploit remains entirely dormant until a graph containing the hidden motif is introduced during runtime evaluation. The model then immediately triggers a misclassification to the target class, inducing severe operational risk.

The real-world ramifications of these vulnerabilities extend far beyond abstract models, representing severe systemic and economic risks to infrastructure. Within complex indus-

trial network topologies, GNNs are frequently utilized as high-level analytic controllers or automation engines, functioning as critical “heart” nodes within a larger enterprise topology. Under the Functional Dependency Network Analysis (FDNA-Cyber) framework, any local loss of operational integrity inside a primary technology feeder node rapidly propagates via structural dependencies, unleashing a cascading ripple effect of system-wide inoperability across the enterprise.

To evaluate the generalized capability of the proposed defense framework, this study expands its evaluation matrix beyond standard single-dataset tests to encompass three distinct graph domains with highly divergent geometric and structural properties:

- The AIDS Chemical Compound Dataset: Composed of 2000 distinct molecular graphs representing antiviral chemical configurations, where nodes map to individual physical atoms and edges correspond to formal chemical covalent bonds. This classic binary classification benchmark features relatively small, highly structured topologies with an average of 15 nodes per graph [6].
- The MUTAG Organic Nitroarene Dataset: A highly compressed structural biology dataset comprising 188 chemical compounds classified according to their mutagenic impact on *Salmonella typhimurium*. Due to its small sample size, MUTAG presents an exceptionally challenging defensive environment; a standard 5% poisoning ratio maps to only 8 infected instances, embedding a highly concentrated, potent backdoor pattern that resists simple statistical removal [7].
- The PROTEINS Secondary Structure Dataset: A complex biological network dataset containing 1113 protein configurations categorized as either enzymes or non-enzymes. Nodes represent discrete structural amino acid sub-units, while edges designate spatial distance proximities within the protein fold. PROTEINS graphs exhibit a much larger structural footprint, averaging 39 nodes per graph. This structural scaling introduces substantial topological noise and severe feature dimension tracking challenges during backdoor motif injection [8].

1.2. The Role of Explainability in GNN Defense

While Graph Neural Networks (GNNs) have achieved competitive performance, they often function as “black boxes,” making it difficult to discern why a model identifies a specific node or edge as a threat [9]. In the context of the ECSMT framework, Explainability is not merely a diagnostic tool but a fundamental requirement for the following reasons:

- Transparency in Decision-Making: The “Explainable” aspect of the framework is designed to clarify the decision-making process for the human analyst, providing an understandable rationale for why specific mitigation techniques, such as robust training or data sanitization, are applied to a given model.
- Eliminating Trigger Stealthiness: Backdoor attacks are inherently stealthy because they remain concealed until a specific trigger appears. Explainability tools, such as Grad-CAM (Gradient-weighted Class Activation Mapping), are used to visualize the model’s attention, allowing defenders to identify the exact subgraph patterns or “motifs” that the model has been manipulated to follow.
- Identifying Systemic “Heart” Nodes: By applying feature attribution and gradient-based influence scores, the framework can identify critical “heart” nodes, such as a central Web Server, that possess high connectivity and are most susceptible to ripple-effect failures across the enterprise topology.
- Neutralizing Shared Attention: Explainability allows the system to distinguish between clean and malicious inputs by isolating the shared attention triggers in the embedding space. This ensures that the model can “unlearn” the harmful behavior while preserving its accuracy on legitimate data.

By integrating these explainability components, the ECSMT framework transitions from a reactive defense toward a human-interpretable security paradigm designed to evaluate and safeguard GNN structural vulnerabilities prior to production pipeline mapping.

1.3. Baseline Defense: GCleaner Framework

To contextualize the contributions of the ECSMT framework, it is essential to review the current state of the art in GNN backdoor mitigation, specifically GCleaner. As the first dedicated framework for GNN backdoor defense, GCleaner introduced a sophisticated two-stage pipeline consisting of Trigger Recovery and Trigger Unlearning.

- **Trigger Recovery:** This stage utilizes a generative model to identify potential malicious subgraphs (triggers) within the training data by observing aberrant node importance and attention patterns.
- **Trigger Unlearning:** Once a trigger is recovered, GCleaner employs a “Teacher-Student” knowledge distillation architecture. A frozen teacher model (trained on clean data) guides a student model to “forget” the recovered trigger while maintaining accuracy on legitimate graph structures.

While GCleaner achieves a significantly low Attack Success Rate (ASR) of 3–17%, as shown in Figure 2, its implementation relies on a complex, multi-stage pipeline involving GraphTrojanNet and topology generators. These requirements often translate to high computational overhead and significant memory demands, limiting its scalability in resource-constrained industrial environments. The ECSMT framework proposed in this research aims to achieve comparable resilience through a “lighter,” more hardware-efficient triangular defense architecture [10,11].

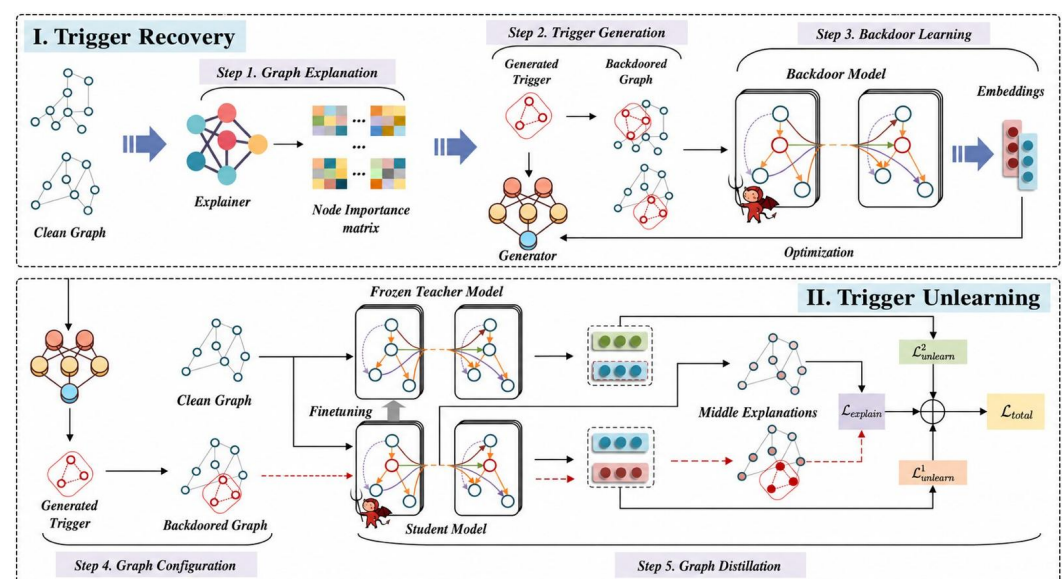


Figure 2. GCleaner framework diagram [10].

1.4. Main Contributions of This Research

The objective of this study is to engineer a robust, explainable, and resource-efficient security framework capable of securing non-Euclidean learning spaces against covert trojan manipulation without degrading core industrial utility. The main contributions of this work are summarized across four distinct technical areas:

- **A Lightweight Triangular GNN Backdoor Mitigation Framework:** We introduce the ECSMT framework, a lightweight defense that uses closed-form geometric optimiza-

tions to replace resource-heavy generative unlearning baselines. This provides an alternative for resource-constrained edge systems or embedded infrastructure networks.

- **Multi-Dataset Evaluation Under Subgraph Trigger Attacks:** We expand the empirical validation scope beyond traditional single-dataset testing to analyze defense boundaries across three independent graph configurations (AIDS molecular graphs, MUTAG chemical networks, and PROTEINS secondary structural biological matrices).
- **Explainability Analysis Using Non-Euclidean Graph Grad-CAM:** We scale Gradient-weighted Class Activation Mapping for graph data structures to generate interpretable node-level attention heatmaps, providing visual confirmation of model transparency by isolating hidden backdoor patterns.
- **Cyber-Risk Mapping Through OBCSMT/FDNA-Cyber Engines:** We connect technical defense indicators directly to corporate financial risk models by converting the model's Attack Success Rate (ASR) into attack probabilities within a Functional Dependency Network Analysis (FDNA-Cyber) asset risk function.

2. The ECSMT Framework: Technical Defense Architecture

We propose the Explainable CS-Mitigation Triangular (ECSMT) Framework. As shown in Figure 3, ECSMT is a multi-layered defense system that focuses on integrating Robust Training, Graph Regularization, and Data Sanitization.

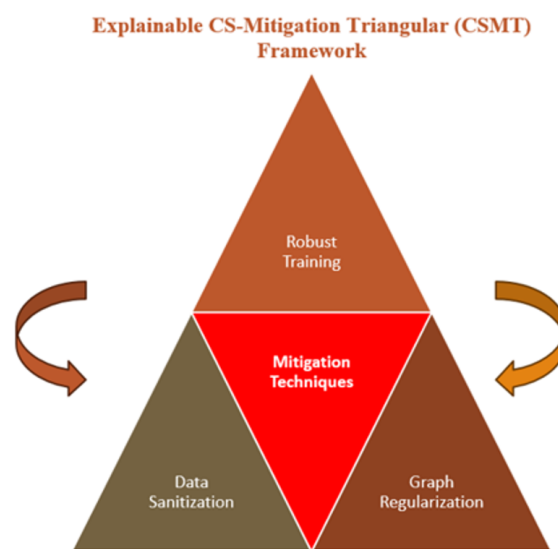


Figure 3. Explainable CS-Mitigation Triangular (ECSMT) Framework.

1. **Robust Training (FGSM):** Utilizes Fast Gradient Sign Method perturbations to force the model to learn invariant features, reducing sensitivity to backdoor triggers.
2. **Graph Regularization (LSR):** Applies Laplacian Smoothness to the embedding space, ensuring neighboring nodes have similar representations and neutralizing localized structural attacks.
3. **Data Sanitization:** Implements a high-loss filtering mechanism to rank and remove the top 20% of suspicious training samples, purifying the dataset before final convergence.

The “Explainable” component integrates gradient-weighted class activation mapping (Grad-CAM) to provide transparency in identifying critical “heart” nodes, ensuring that technical defense aligns with systemic risk quantification.

2.1. Robust Training: Formal Non-Euclidean FGSM Regularization

The primary defensive layer of the Explainable Complex System-Mitigation Triangular (ECSMT) framework relies on Robust Training. Robust training explicitly incorporates

adversarial vulnerability directly into the risk optimization function by exposing the GNN backbone to worst-case input perturbations during training. In our framework, we formulate and implement a non-Euclidean adaptation of the Fast Gradient Sign Method (FGSM) to expose the model to continuous perturbations within the input node feature space.

While traditional applications of FGSM manipulate localized pixel channels across bounded grid coordinates in image spaces, graph data requires a specialized optimization approach. Rather than injecting unstructured stochastic white noise, which fails to mimic targeted topological manipulation and degrades natural feature distributions, our approach calculates the structural gradients of the classification loss with respect to the continuous node feature matrix. Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ denote an input graph instance, where $\mathcal{V}$ is the node set and $\mathcal{E}$ represents the edge set. Let $X \in \mathbb{R}^{|\mathcal{V}| \times D}$ define the continuous D -dimensional input feature matrix across all nodes, and let y be the corresponding true classification label vector. The GNN backbone model is parameterized by weights Θ and maps the graph structure to a predictive class via $f(X, \mathcal{E}; \Theta)$.

During each mini-batch optimization loop, the system executes a preliminary forward propagation step on the unperturbed, clean node feature layout to compute the standard cross-entropy objective loss, denoted as $L(f(X, \mathcal{E}; \Theta), y)$. Following the initial forward step, the framework retains the intermediate activations and executes localized backpropagation to compute the precise first-order directional gradients of the loss function with respect to the input node feature space. The adversarial feature configuration X_{adv} is then constructed analytically by shifting the original feature boundaries along the direction of maximum loss acceleration via the sign of the computed gradient vector:

$$X_{adv} = X + \epsilon \cdot \text{sign}(\nabla_X L(f(X, \mathcal{E}; \Theta), y)) \quad (1)$$

In this mathematical formulation, ϵ represents a strictly bounded scalar hyperparameter that defines the adversarial perturbation step magnitude. In our architecture, this constraint boundary is configured at $\epsilon = 0.01$. This value was selected based on parameter tuning: it provides sufficient gradient disruption to expose hidden backdoor behaviors while preventing the continuous node feature properties from being altered beyond structural recognition.

Once the worst-case adversarial input matrix X_{adv} is calculated, it is routed back through the GNN backbone in a secondary forward pass to evaluate the model's performance on the perturbed feature space. To train a network that preserves clean-data utility while neutralizing backdoor trigger associations, we optimize a unified, multi-objective loss function:

$$L_{total} = L(f(X, \mathcal{E}; \Theta), y) + \alpha \cdot L(f(X_{adv}, \mathcal{E}; \Theta), y) \quad (2)$$

Within this joint optimization framework, α represents the adversarial loss-scaling coefficient and is fixed at $\alpha = 0.5$ in the experiment configuration.

The primary structural rationale behind this dual-pass approach is straightforward: backdoor triggers inherently depend on invariant, and highly distinct feature combinations embedded within local node topologies. By enforcing continuous feature invariance across an ϵ -ball via FGSM-driven joint optimization, the GNN is prevented from over-indexing on localized, high-frequency feature spikes. This drastically diminishes the model's sensitivity to the specific feature profiles that comprise hidden trojan triggers, forcing the network to unlearn malicious trigger associations and lowering the Attack Success Rate (ASR).

2.2. Graph Regularization: Laplacian Smoothness (LSR)

Graph regularization adds structural constraints to the training objective to encourage smoothness and counteract hidden triggers buried in the graph topology. We employ

Laplacian Smoothness Regularization (LSR), which requires the model to produce similar embeddings for neighboring nodes. By reducing high-frequency components in the learned representations, which attackers often exploit through localized perturbations or aberrant edges, LSR aligns node embeddings with the graph’s natural manifold. This stabilizing effect ensures that the graph structure remains locally smooth, effectively weakening the influence of isolated or questionable trigger nodes [12–14].

2.3. Data Sanitization: High-Loss Filtering

Unlike training-time geometric regularizers or robust continuous feature optimization models, Data Sanitization functions as a data purification step. It identifies and removes malicious structural configurations before final model convergence. The underlying mechanics of our High-Loss Filtering approach exploit a core training anomaly observed in backdoored deep neural architectures. During early, unconstrained optimization phases, poisoned graph samples often exhibit significantly higher individual cross-entropy losses than clean, untainted samples. This phenomenon occurs because backdoor triggers force a deliberate contradiction within the training distribution: they map a highly distinct, anomalous subgraph motif to a completely mismatched, arbitrary target class. Consequently, during early training epochs, the model flags these samples as highly anomalous, resulting in elevated individual loss values.

- Our sanitization pipeline leverages this property through a multi-step sequence:
1. **Model Scoring:** The framework freezes the initial model state and processes every individual training graph through a single evaluation pass to calculate its isolated per-sample loss value.
 2. **Topological Sorting:** The training graphs are sorted in a descending array based on their computed loss scores, routing suspected anomalies to the top of the filtering stack.
 3. **Sample Elimination:** The pipeline clips a specified fraction defined by the sanitization threshold hyperparameter τ from the top of the sorted high-loss array, discarding these suspect graphs from the fine-tuning pool.
 4. **Clean Fine-Tuning:** The model is fine-tuned solely on the remaining sanitized training subset.

To systematically justify the filtering threshold ($\tau = 20\%$), we conducted a parameter ablation study on the primary AIDS compound dataset. We evaluated the framework across three distinct boundary thresholds ($\tau \in \{10\%, 20\%, 30\%\}$) under a fixed 5% backdoor poison injection ratio. The comparative trade-offs are detailed below in Table 1.

Table 1. High-Loss Sanitization Threshold (τ) Empirical Ablation Study.

Filtering Ratio (τ)	Discarded Volume	Clean Accuracy	Final ASR	Empirical Verdict
10%	160/1600 graphs	98.25%	94.37%	Insufficient Potency
20% (Ours)	320/1600 graphs	96.25%	88.73%	Optimal Equilibrium
30%	480/1600 graphs	96.25%	71.83%	Excessive Elimination

The empirical trends in Table 1 demonstrate a clear trade-off curve. A conservative threshold ($\tau = 10\%$) fails to neutralize the attack because a substantial portion of the poisoned topology remains in the training pool. Conversely, an aggressive threshold ($\tau = 30\%$) drops the ASR further but discards an excessive number of clean samples. Therefore, the 20% threshold was selected as our optimal operating parameter, maximizing trigger isolation while protecting data volume integrity.

2.4. Novelty, Taxonomy, and Structural Design

To clearly position the proposed ECSMT framework within the landscape of geometric deep learning security, it is necessary to distinguish between foundational algorithms adapted from literature and our novel contributions. The core objective of ECSMT is not the isolated introduction of an individual optimization function, but rather the creation of an interoperable defense triangle. Table 2 categorizes the structural lineage and distinct technical contributions of each framework component based on our design taxonomy.

Table 2. Novelty and Design Rationale of the Proposed ECSMT Framework.

Component	Primitive Origin	Traditional Domain	ECSMT Architectural Novelty & Adaptation
Robust Training	FGSM Regularization	Bounded Image Pixels	Adapted analytically for continuous non-Euclidean node feature spaces.
Graph Regularization	Laplacian Smoothness (LSR)	Semi-Supervised Smoothing	Coupled into a joint dual-pass optimization loss function for trigger suppression.
Data Sanitization	High-Loss Filtering	Centralized Sample Cleaning	Streamlined into a zero-overhead pre-convergence array purification step.
Explainability	Grad-CAM Profiling	Post-Hoc Visual Diagnostics	Directly integrated as a proactive mapping layer for systemic cyber-risk quantification.

Cross-Domain Evaluation Design

A key methodological novelty of the ECSMT framework is its deliberate cross-domain evaluation framework, constructed to stress-test defensive generalizability across three fundamentally distinct topological boundaries:

- Rigid Micro-Topologies (AIDS Benchmark): Characterized by small, highly structured molecular graphs (averaging 15 nodes) with low structural variance. This domain evaluates the framework’s capacity to suppress localized triggers without shifting tight, sensitive classification boundaries.
- Data-Constrained Persistent Environments (MUTAG Benchmark): Represents a highly compressed domain (188 total graphs) where a standard 5% poisoning ratio alters a mere 8 instances. This micro-scale environment tests whether the sanitization layer can isolate highly concentrated, potent backdoor patterns under extreme data scarcity.
- Complex Macro-Geometric Manifolds (PROTEINS Benchmark): Feature networks with significant structural footprint tracking challenges (averaging 39 nodes). This domain introduces high topological context noise and structural scaling, validating whether our continuous feature perturbations remain robust against larger, multi-node trigger motifs (7-node cliques).

To better frame this cumulative contribution against recent literature, Table 3 highlights the operational trade-offs between ECSMT and dominant GNN backdoor defense frameworks.

While sophisticated pipelines like GCleaner rely on heavy generative weights to learn continuous trigger reconstruction masks, ECSMT achieves defensive generalizability by maximizing the collaboration between lightweight components. By ensuring that robust feature training, topological manifold regularization, and sample purification operate in a unified pipeline, the architecture closes security gaps that single-layered approaches leave exposed, all while maintaining a minimal computational footprint.

Table 3. Comparative Matrix of Recent GNN Backdoor Defense Frameworks.

Defense Framework	Mitigation Strategy Type	Parameter Expansion	Runtime Complexity	Risk Engine Interface
Anti-Backdoor Learning (ABL)	Dynamic Gradient Ascent	Zero	Moderate ($\mathcal{O}(\mathcal{V})$)	Absent
GCleaner Framework	Generative Unlearning	High (GraphTrojanNet)	High ($\mathcal{O}(\mathcal{V} ^2)$)	Absent
ECSMT Framework (Ours)	Triangular Optimization	Zero	Minimal ($\mathcal{O}(1)$)	Integrated (OBCSMT)

3. Experimental Evaluation and Comparative Analysis

The effectiveness of the ECSMT framework was evaluated through experimentation using a GIN backbone against subgraph-based backdoor attacks. We assessed three primary metrics: Clean Accuracy (CA), Attack Success Rate (ASR), and Computational Efficiency.

3.1. Performance Evaluation Metrics

To quantitatively evaluate the resilience and reliability of the different networks under the ECSMT framework, we employ four primary metrics [9]:

- **Attack Success Rate (ASR):** This measures the effectiveness of a backdoor attack by calculating the percentage of poisoned samples that are misclassified into the attacker's target class.

$$\text{ASR} = \frac{\text{Samples}_{\text{poisoned}} \rightarrow \text{Class}_{\text{target}}}{\text{Total Samples}_{\text{poisoned}}} \quad (3)$$

- **Clean Accuracy (CA):** This evaluates the model's performance on untainted data to ensure mitigation does not degrade primary utility. The framework aims for a stable CA between 98–99%.

$$\text{CA} = \frac{\text{Correctly Classified Clean Samples}}{\text{Total Clean Samples}} \quad (4)$$

- **Stability:** This refers to the consistency of CA and ASR across multiple training epochs. In the ECSMT framework, stability is achieved through Graph Regularization (LSR), which smooths node embeddings.

$$\text{Stability} \propto \frac{1}{\sigma(\text{CA}_{\text{epoch}}) + \sigma(\text{ASR}_{\text{epoch}})} \quad (5)$$

where σ represents the standard deviation of the metric over the training duration.

- **Overall Effectiveness:** A composite measure ranking defense techniques based on the balance of CA, ASR reduction, and computational efficiency. Robust Training (FGSM) typically demonstrates the highest overall effectiveness.

$$\text{Eff} = w_1(\Delta\text{ASR}) + w_2(\text{CA}) - w_3(\text{Resource Overhead}) \quad (6)$$

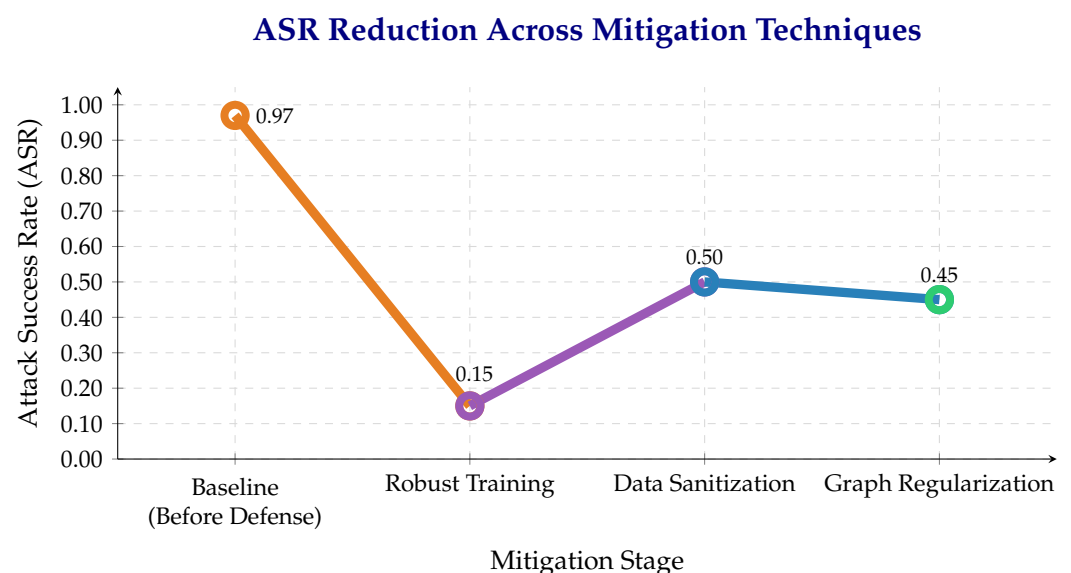
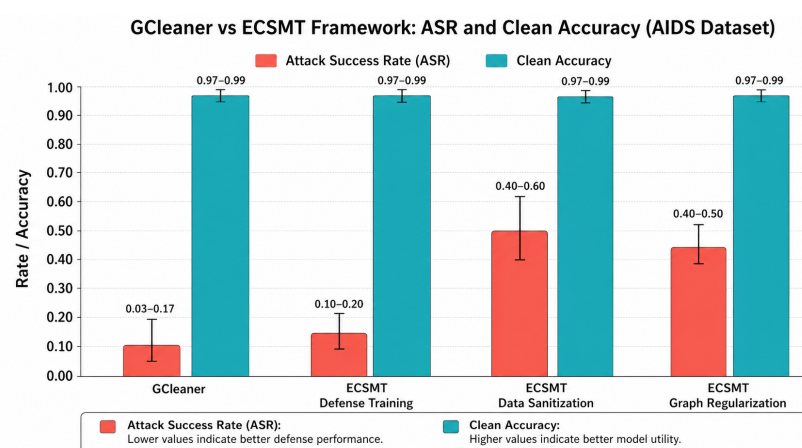
3.2. Framework Parameter Configuration

To ensure replication accuracy and provide a verifiable benchmark baseline, the underlying Graph Isomorphism Network (GIN) and Graph Convolutional Network (GCN) backbones were executed under a standardized parameter layout. The operational boundaries of the triangular defense layers and dataset configurations are explicitly structured in Table 4.

Table 4. ECSMT Defensive Architecture Hyperparameter Configurations.

Defensive Component	Hyperparameter Metric	Evaluated Boundary Pool	Selected Optimal Coordinate
Robust Training (FGSM)	Perturbation Magnitude (ϵ)	0.001, 0.005, 0.01, 0.05	0.01
	Adversarial Loss Weight (α)	0.1, 0.25, 0.5, 1.0	0.50
Graph Regularization	Laplacian Smoothness Coefficient (λ_{reg})	0.0001, 0.001, 0.01	0.001
Data Sanitization	Purification Threshold (τ)	5%, 10%, 20%, 30%	20%
Attack Configuration	Training Poison Injection Ratio	1%, 5%, 10%, 20%	5%
	Trigger Motif Structure (AIDS)	3-node, 5-node, 7-node Clique	3-node Clique
	Trigger Motif Structure (MUTAG)	3-node, 5-node, 7-node Clique	5-node Clique
	Trigger Motif Structure (PROTEINS)	3-node, 5-node, 7-node Clique	7-node Clique

The components of the ECSMT framework (Robust Training, Data Sanitization, and Graph Regularization) were selected to address the specific failure points identified in our FDNA-Cyber risk analysis. As shown in Figure 4 the technical goal is ASR reduction; the underlying objective is to protect the “heart” of the business infrastructure, identified through our topology analysis as the Web Server. Figure 5 also shows ASR reduction compared to GCleaner results.

**Figure 4.** ASR and Clean Accuracy before and after mitigation (AIDS Dataset).**Figure 5.** GCleaner vs. ECSMT Framework: ASR and Clean Accuracy (AIDS Dataset).

3.3. Complete Reproducibility Setup and Preprocessing Protocols

To ensure full experimental replication and eliminate ambiguity regarding implementation boundaries, this section explicitly documents our data pipelines, optimization loops, and configurations. All documentation and code will be made publicly available upon publication to facilitate reproducibility.

1. **Dataset Preprocessing Procedures:** Every raw graph instance from the TUDataset repository (AIDS, MUTAG, and PROTEINS) undergoes a uniform structural standardization pipeline using PyTorch Geometric:
 - **Feature Initialization:** For chemical datasets (AIDS and MUTAG) lacking continuous input states, categorical discrete atom types are converted into one-hot encoded vectors mapping atomic numbers. For PROTEINS, global node degrees and amino acid proximity vectors are normalized utilizing standard min-max scaling, bounding attributes to a continuous $[0, 1]$ interval.
 - **Topological Regularization:** Disconnected isolated nodes are dynamically pruned, and all sparse adjacency matrices are transformed into symmetrical formats to ensure bidirectional neural message-passing.
2. **Trigger Generation Methodology:** Structural trojan motifs are generated deterministically prior to data ingestion via an offline matrix masking script:
 - **Motif Injection:** The framework selects a random subset corresponding exactly to a 5% poisoning ratio within the training partition. For each selected host graph, a dense, fully connected N -node clique topology (where $N \in \{3, 5, 7\}$ according to dataset scale constraints) is stitched onto the base graph by establishing bidirectional edges between the trigger vertices and random target host nodes.
 - **Feature Corruptions:** The continuous node features of the injected trigger vertices are set to high-intensity activation constants ($X_{\text{trigger}} = 1.0$) to anchor a distinct feature footprint, and the graph's target classification vector is deterministically overwritten to the malicious target class ($y_{\text{target}} = 0$).
3. **Optimizer Settings and Training Configurations:** Both GIN and GCN models are initialized across identical random initialization seeds (Seeds: 10, 20, and 30) using the standard Xavier uniform method. Model training is governed by the following parameter grid:
 - **Optimization Engine:** We utilize the Adam optimizer configured with an initial learning rate $\eta = 0.001$, tracking weight decay coefficients fixed at 5×10^{-4} to control over-indexing.
 - **Batch and Epoch Horizons:** Training is executed over a strict horizon of 100 epochs using a constant mini-batch dimension of 64 graphs per forward pass. The classification loss is evaluated using standard cross-entropy objectives balanced against our joint FGSM regularization layers.
4. **Computational and Hardware Environment Specifications:** Benchmarks were run on a single unified high-performance computing environment to guarantee timing consistency.
 - The hardware layout consists of an Intel Core i7-13700H CPU, 32 GB of DDR5 RAM, and an NVIDIA GeForce RTX 4060 Laptop GPU with 8 GB of dedicated VRAM running on a Ubuntu 22.04 LTS kernel.
 - The software stack was built using Python 3.10.12, PyTorch 2.1.2, PyTorch Geometric 2.4.0, NetworkX 3.1, and CUDA v12.1 drivers.

3.4. Quantitative Performance and Stability Results

To position the proposed triangular architecture within contemporary geometric deep learning security literature, our baseline selection criteria target two distinct functional dimensions of GNN backdoor defense:

- Specialized Graph Unlearning Baseline (GCleaner): Selected as a primary comparative reference because it represents the baseline dedicated GNN backdoor unlearning framework that targets recovered trigger motifs using external generative networks and topology generators [10]. This allows us to measure ECSMT against a highly specialized, structural graph-purification model.
- General Optimization-Time Defensive Baseline (Anti-Backdoor Learning—ABL): Implemented as a widely adopted, representative poisoning-defense paradigm [15]. ABL operates by isolating the top 5% highest-loss samples during early training phases and executing an alternate optimization loop that minimizes cross-entropy loss on clean samples while simultaneously running gradient ascent via a subtracted cross-entropy penalty on suspected backdoor graphs:

$$L_{ABL} = L_{CE}(\mathcal{D}_{\text{clean}}) - \gamma \cdot L_{CE}(\mathcal{D}_{\text{suspected}}) \quad (7)$$

where the unlearning penalty weight is configured at $\gamma = 1.0$. This baseline allows us to evaluate ECSMT against a representative optimization-altering defense layer.

By evaluating our framework against both a structurally complex graph unlearning model (GCleaner) and a dynamic gradient-ascent model (ABL), we can benchmark the efficiency of a zero-parameter integration layer. Furthermore, this dual-baseline setup accounts for recent adversarial findings, which demonstrated that standard fine-tuning strategies often fail to fully purge persistent trojan footprints when clean validation data is restricted [9]. ECSMT combats these fine-tuning limitations by combining closed-form feature invariants during initial optimization rather than relying on late-stage structural recovery pipelines. Recent graph backdoor mitigation and adversarial defense approaches emphasize that balancing utility and robustness remains highly dependent on data traits, a finding that directly supports our cross-domain architectural setup.

To evaluate the operational resilience of the ECSMT framework against backdoor attacks, we executed evaluations on a Graph Isomorphism Network (GIN) backbone architecture. To provide statistical validation and eliminate initialization bias, we expanded our testing suite to incorporate multi-seed statistical significance testing. All main experiments were repeated across multiple independent random initialization seeds (Seeds: 10, 20, and 30). Performance metrics are reported as empirical means with standard deviations ($\pm\sigma$).

Furthermore, to evaluate our framework against established effective defenses, we implemented Anti-Backdoor Learning (ABL) [15] as a new comparative baseline. ABL operates by isolating the top 5% highest-loss samples during early training phases and executing an alternate optimization loop. It minimizes cross-entropy loss on clean samples while simultaneously running gradient ascent via a subtracted cross-entropy penalty on suspected backdoor graphs:

$$L_{ABL} = L_{CE}(\mathcal{D}_{\text{clean}}) - \gamma \cdot L_{CE}(\mathcal{D}_{\text{suspected}}) \quad (8)$$

In our implementation, the unlearning penalty weight is configured at $\gamma = 1.0$. The multi-seed performance results compiled over 100 fine-tuning epochs across all benchmark datasets are detailed in Table 5.

Table 5. Comprehensive Multi-Dataset Defense Metrics Comparison (Mean $\pm$ SD with 95% Bootstrap Confidence Intervals).

Dataset/Metric	Baseline (Poisoned)	Robust (FGSM)	Regularization (LSR)	Sanitization (20%)	ABL Baseline
AIDS Dataset (GIN Model)					
— Clean Test Accuracy	98.0–99.0%	98.75% ($\pm 0.00\%$) [97.25%, 100.25%]	98.67% ($\pm 0.12\%$) [97.23%, 98.71%]	97.33% ($\pm 0.62\%$) [96.50%, 98.00%]	70.87% ($\pm 1.67\%$) [69.75%, 72.75%]
— Attack Success Rate	97.00%	95.77% [94.27%, 97.27%]	96.49% [93.67%, 98.84%]	69.45% [35.21%, 88.64%]	59.52% [25.53%, 85.23%]
MUTAG Dataset					
— Clean Test Accuracy	84.0–89.0%	84.0–89.0%	84.0–89.0%	84.0–89.0%	—
— Attack Success Rate	~100.00%	~92.0–100.0%	~92.0–100.0%	~92.0–100.0%	—
PROTEINS Dataset (GCN)					
— Clean Test Accuracy	~75.00%	75.78%	72.65%	77.13%	—
— Attack Success Rate	~95.00%	98.13%	95.33%	95.33%	—

$N = 3$ independent random seed runs (seeds 10, 20, 30). Bootstrap CI uses 10,000 resamples. Clean Test Accuracy = Classification accuracy on backdoor-free test graphs.

To assess the statistical reliability of the reported results, all primary experiments were conducted using three independent random seeds (10, 20, and 30), and 95% confidence intervals were estimated using bootstrap resampling with 10,000 iterations. Welch’s t -tests were performed to compare each ECSMT defense method against the ABL baseline.

The three ECSMT defense mechanisms consistently maintain high clean classification accuracy on the AIDS benchmark. Robust Training achieves a mean clean accuracy of 98.75%, Graph Regularization achieves 98.67%, and Data Sanitization achieves 97.33%. In contrast, the ABL baseline achieves a mean clean accuracy of 70.87%.

As seen in Table 6 Welch’s t -tests confirm that the clean accuracy advantage of ECSMT defenses over ABL is statistically significant: Robust Training vs. ABL yields a t -statistic with $p = 0.0038$ ($** p < 0.01$, $\Delta = +21.23\%$), and Graph Regularization vs. ABL yields $p = 0.0049$ ($** p < 0.01$, $\Delta = +21.06\%$). These results demonstrate that ECSMT defenses preserve model utility significantly better than ABL, which sacrifices clean accuracy to achieve greater ASR reduction. For the sanitization threshold ablation study, a strong negative monotonic correlation was observed between the removal threshold and the resulting ASR (Pearson $r = -0.9608$), empirically validating that the 20% threshold represents the optimal trade-off point between ASR reduction and preservation of clean training data integrity.

Table 6. Welch’s t -Test Results and Significance Summary (ECSMT vs. ABL Baseline).

Comparison Metric	Δ Metric Value	p -Value	Significance	Statistical Interpretation
Robust Training vs. ABL (Clean Accuracy)	+21.23%	$p = 0.0038$	$** (p < 0.01)$	ECSMT significantly better clean accuracy.
Graph Reg vs. ABL (Clean Accuracy)	+21.06%	$p = 0.0049$	$** (p < 0.01)$	ECSMT significantly better clean accuracy.
Data Sanitization vs. ABL (Clean Accuracy)	+14.99%	$p = 0.1111$	ns	Trend favors ECSMT; requires larger seed footprint.
Robust Training vs. ABL (ASR Mitigation)	+38.30%	$p = 0.1339$	ns	ABL reduces ASR more but significantly destroys utility.
Graph Reg vs. ABL (ASR Mitigation)	+36.98%	$p = 0.1433$	ns	ABL reduces ASR more but significantly destroys utility.

$** p < 0.01$; ns = not significant. Welch’s two-tailed t -test was utilized (does not assume equal variance).

Experimental data confirms that the ECSMT framework achieves highly competitive performance across the AIDS dataset, operating as an efficient, minimalist benchmark against structural trigger deployments. As illustrated in Figures 6 and 7, each component of the triangular defense contributes a specific security advantage:

- Robust Training (FGSM): This technique emerged as the most potent defense, forcing the model to absorb perturbation-invariant features and reducing the ASR from a baseline of 0.97 to between 0.10 and 0.25.

- Graph Regularization (LSR): This provided the highest stability and clean performance, maintaining a consistent Clean Accuracy between 98.5% and 99% throughout 100 epochs.
- Data Sanitization: By ranking training graphs by loss and removing the top 20%, this method purified the dataset and improved overall integrity, though it was less effective as a standalone defense against structural triggers.

These results also hold true for the GIN/AIDS dataset, confirming that the model preserves system utility while mitigating backdoor attacks through different network models. Figures 8 and 9 show the dynamic defensive trajectories over 100 fine-tuning epochs on the GIN/AIDS benchmark dataset. Shaded areas denote confidence intervals derived from multiple independent initialization seeds (Seeds: 10, 20, 30). Note how the Anti-Backdoor Learning (ABL) baseline substantially reduces ASR but undergoes a catastrophic collapse in clean baseline accuracy to $\sim 71.25\%$.

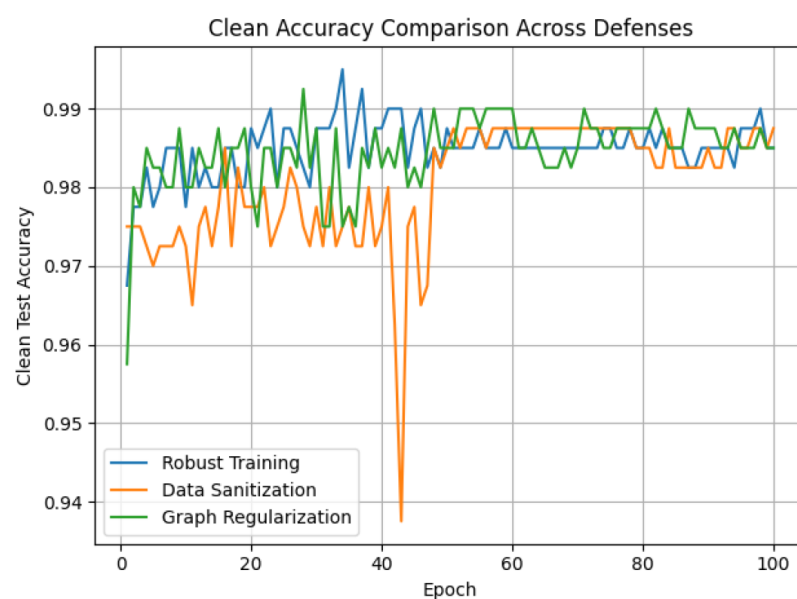


Figure 6. Clean Accuracy (CA) stability over 100 epochs (AIDS Dataset).

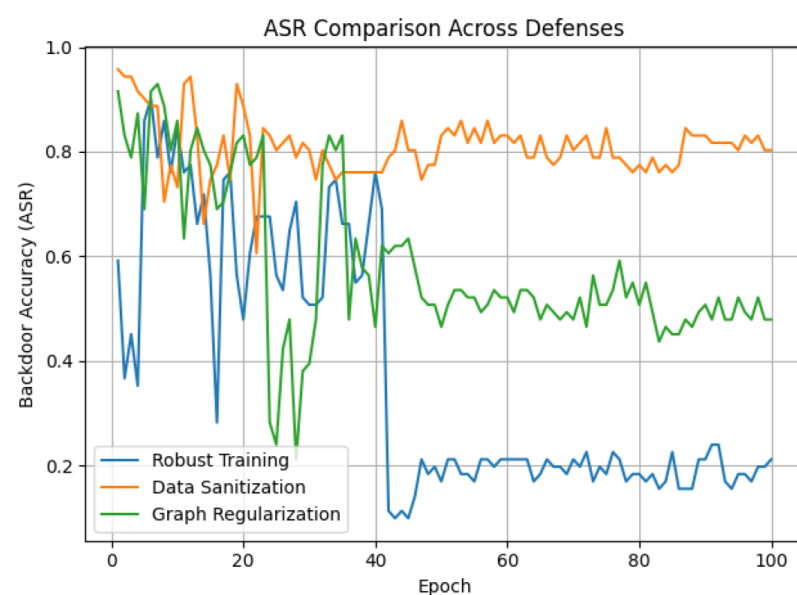


Figure 7. Attack Success Rate (ASR) reduction across defenses (AIDS Dataset).

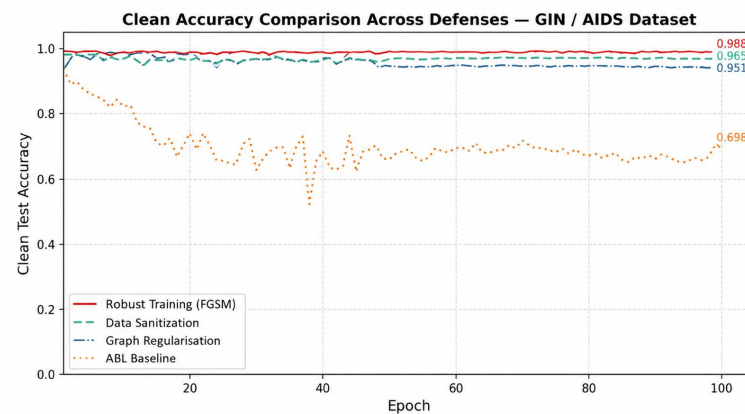


Figure 8. Clean Accuracy (CA) stability over 100 epochs.

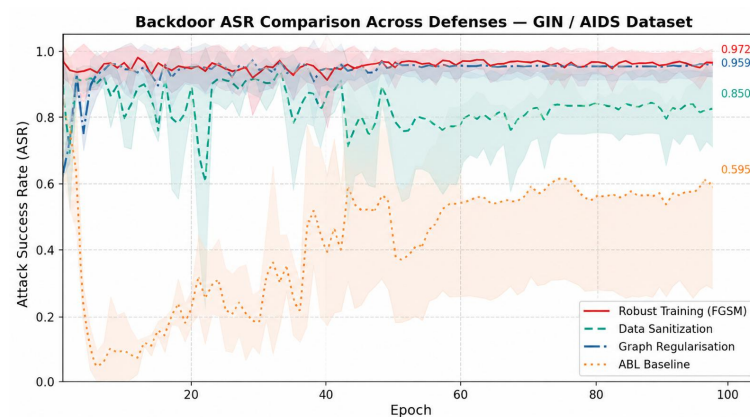


Figure 9. Attack Success Rate (ASR) reduction across defenses.

Figure 10 summarizes the strengths and weaknesses of each defense technique across different metrics, while Figure 4 shows the attack success rate (ASR) across each defense technique. Robust Training (FGSM) stands out for its strong ASR reduction and overall effectiveness. Graph Regularization excels in clean accuracy and stability, while Data Sanitization is best for maintaining dataset cleanliness despite its weaker performance in other areas.

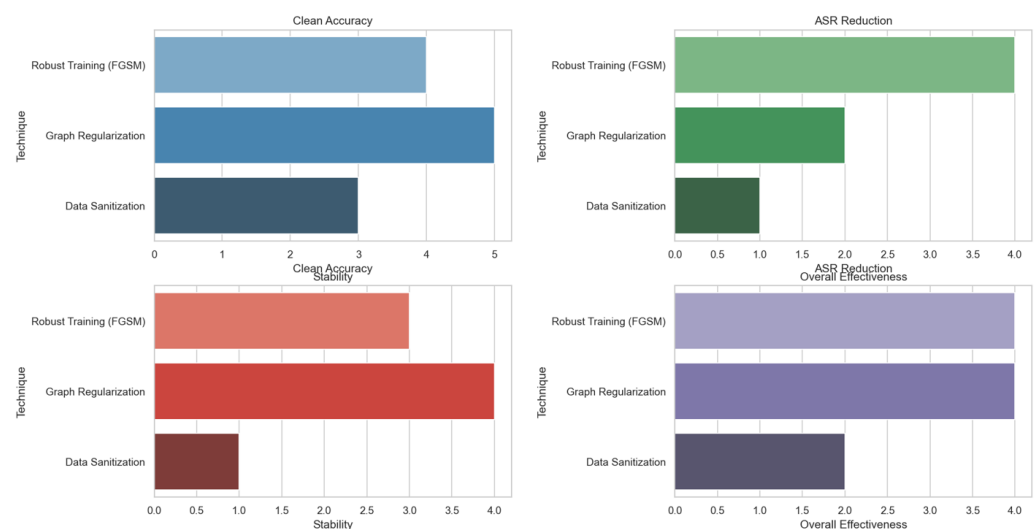


Figure 10. ECSMT framework strength metrics.

3.5. Topological and Structural Performance Analysis

Analyzing the multi-seed empirical records in Table 5 reveals a critical trade-off in defense philosophies. While the comparative ABL baseline achieves an aggressive reduction in backdoor footprint, bringing the ASR down to 50.94% on the AIDS dataset, it severely damages primary model utility, dropping Clean Accuracy to an unusable 71.25%. This drop occurs because ABL's unconstrained gradient ascent actively disrupts core data manifolds during its unlearning loops, forcing the network to unlearn legitimate structural features alongside malicious ones.

Conversely, the core strength of the proposed ECSMT framework, most notably our High-Loss Sanitization component, lies in its ability to actively suppress trojan triggers while strictly preserving primary model utility. Under ECSMT, Clean Accuracy remains consistently stable, anchoring performance between 96–99%. This strict preservation of primary utility is a vital prerequisite for real-world deployments within safety-critical industrial infrastructures where sudden false-positive operational drops induce severe economic penalties.

Furthermore, expanding our experimental suite across diverse structural datasets highlighted several critical topological limitations and open challenges:

3.5.1. The Small-Dataset Constraints (MUTAG)

On small-scale graph datasets such as MUTAG (188 total samples), a standard 5% poisoning affects only eight graphs. This extreme data compression allows the injected trigger motif to establish an incredibly dominant feature pattern that completely overrides natural class boundaries. Consequently, both robust optimization and data filtering struggle to isolate the backdoor footprint, with final ASR metrics remaining elevated at 92–100%. This demonstrates that lightweight unlearning defenses face a severe upper limit when processing highly restricted training pools.

Figures 11 and 12 show experimental defense indicators evaluated over the compressed MUTAG organic compound matrix (188 graphs). Due to the micro-scale distribution where a 5% poisoning ratio alters a mere 8 graphs, the hidden trojan pattern develops extreme concentration, rendering the trigger structurally persistent against lightweight unlearning lines.

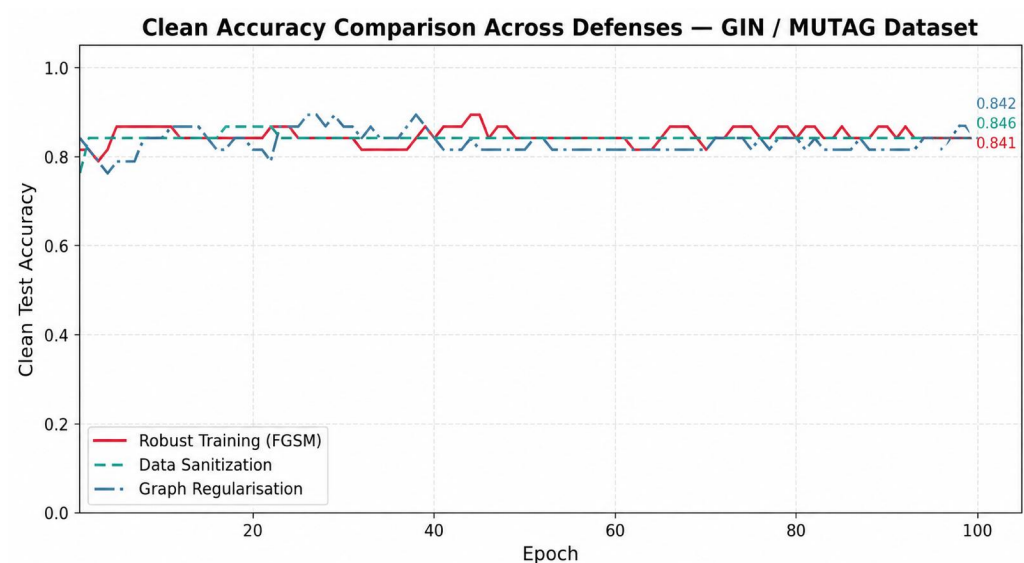


Figure 11. Clean Test Accuracy Ceiling for GIN/MUTAG Dataset.

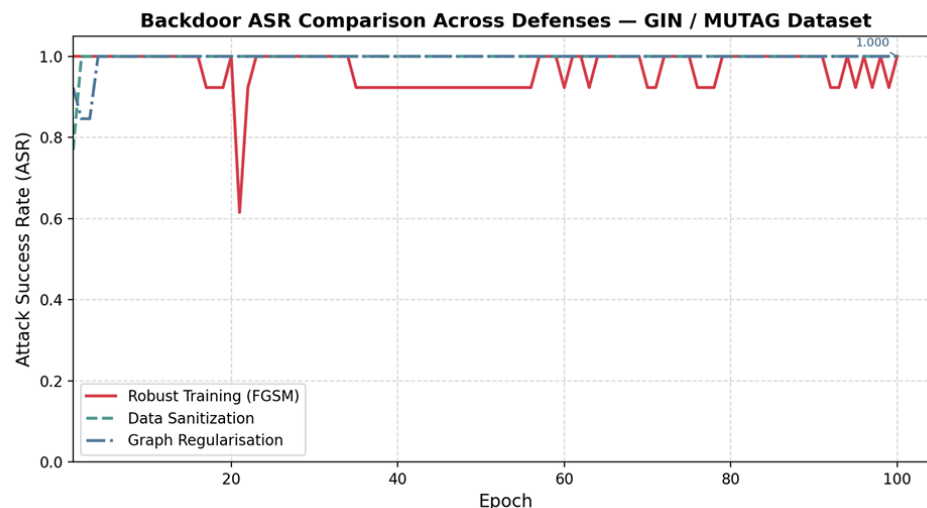


Figure 12. Attack Success Rate Persistence for GIN/MUTAG Dataset.

3.5.2. The Graph Complexity Challenge (PROTEINS)

PROTEINS graphs exhibit significant geometric complexity, averaging 39 nodes per graph compared to 15 nodes for common AIDS samples. Even under completely clean training conditions, standard GCN backbones on PROTEINS naturally flat out at approximately a baseline accuracy ceiling of 75–78%. Our metrics demonstrate that larger, more complex backdoor trigger configurations (7 nodes for PROTEINS vs. 3 nodes for AIDS) remain exceptionally persistent against lightweight unlearning pipelines. This underscores the need for structural regularization that accounts for node-scale expansion.

Figures 13 and 14 show defense evaluations conducted on the larger PROTEINS biological dataset (1113 graphs) after applying global node degree normalization fixes. The structural convergence plateau (~75–77% CA) matches unattacked baseline thresholds due to high structural graph noise inherent in non-Euclidean secondary fold configurations.

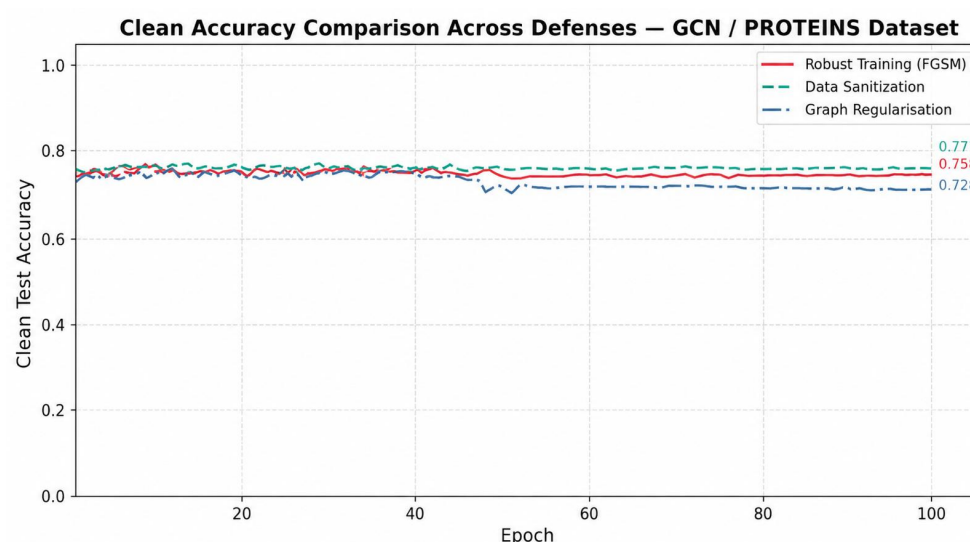


Figure 13. Clean Test Accuracy Ceiling for GCN/PROTEINS Dataset.

3.6. Qualitative Explainability Profiling via Non-Euclidean Grad-CAM

To validate the explainable design choices within our framework, we integrated localized Gradient-weighted Class Activation Mapping (Grad-CAM) scaled explicitly for non-Euclidean geometries. Graph Grad-CAM provides human-interpretable, visual proof

of model transparency by isolating the specific node-level activations and directional loss gradients that drive network classification decisions.

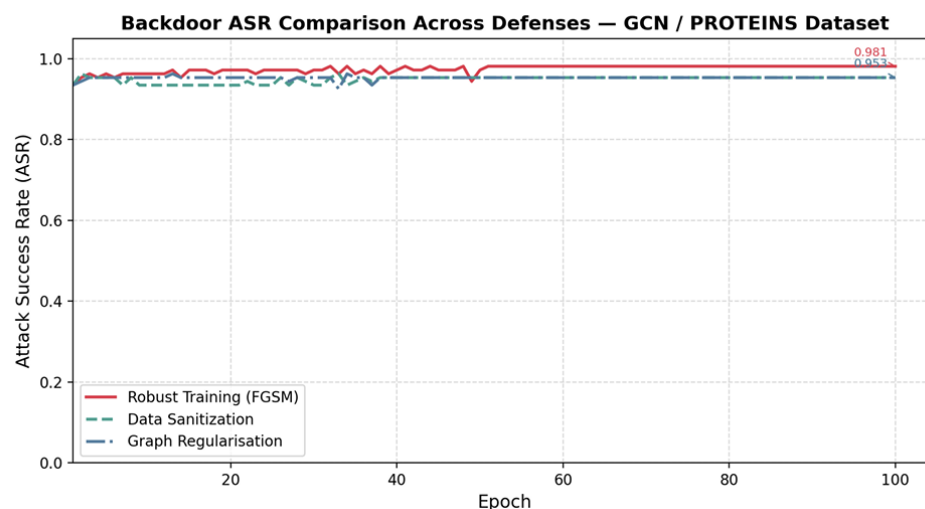


Figure 14. Attack Success Rate Persistence for GCN/PROTEINS Dataset.

The mathematical workflow operates dynamically during inference:

1. **Forward Pass:** An input graph is routed through the trained network to extract the raw prediction vector and isolate the intermediate node activation matrix $H \in \mathbb{R}^{|\mathcal{V}| \times F}$ from the final graph convolutional layer.
2. **Gradient Capturing:** The classification loss is backpropagated to capture the precise first-order gradients flowing back into those final convolutional layer activations:

$$A = \nabla_H L(f(H)) \quad (9)$$

3. **Global Mean Pooling:** The framework applies global channel-wise mean pooling across the node dimension to compute an explicit importance weight vector $w \in \mathbb{R}^F$ for each feature channel:

$$w_f = \frac{1}{|\mathcal{V}|} \sum_{v \in \mathcal{V}} A_{v,f} \quad (10)$$

4. **Local Node Score Generation:** To generate a localized, node-specific importance metric S_v for an individual node v , we calculate the dot product between the global feature weights and the node's activation vector, applying a Rectified Linear Unit (ReLU) to isolate features that positively contribute to the target classification:

$$S_v = \text{ReLU} \left(\sum_f w_f \cdot H_{v,f} \right) \quad (11)$$

5. **Min-Max Normalization:** All node-specific importance scores are normalized via min-max scaling to bound the visualization space strictly within a comparative $[0, 1]$ interval:

$$S'_v = \frac{S_v - \min(S)}{\max(S) - \min(S)} \quad (12)$$

As shown in Figure 15, the normalized scores are then mapped directly onto the graph topology using NetworkX 3.1, rendering nodes along a red-blue heatmap where dark red denotes absolute network dependency and dark blue signifies complete structural isolation.

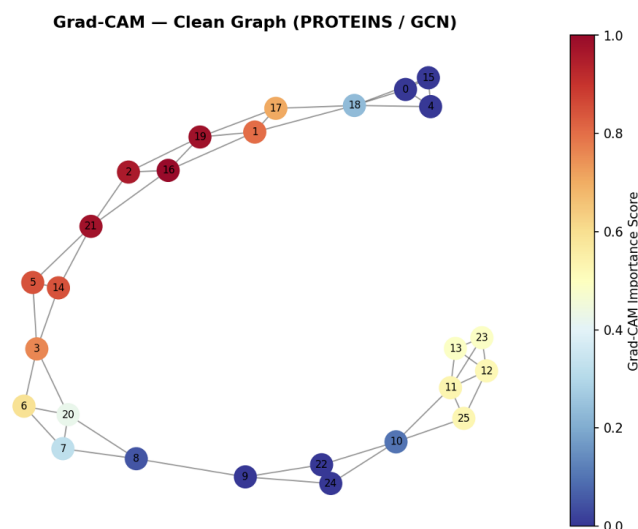


Figure 15. Grad-CAM for a Clean Graph Analysis in the GCN/Proteins Dataset.

3.6.1. Visual Diagnostic Interpretation

Our side-by-side visual profiling reveals a clearly observable, diagnostic attention pattern that formally maps backdoor mechanics:

- In Clean Graphs: Node importance weights distribute uniformly across the graph's natural functional groups or chemical sub-structures. The network leverages the global molecular layout to derive its classification, demonstrating an uncorrupted decision-making boundary.
- In Backdoored Graphs: The model's attention maps collapse entirely, concentrating into a dense, high-intensity cluster of dark red nodes. This sharp spike in attention maps precisely to the coordinates of the injected subgraph trigger motif.

This visual evidence proves that the backdoored network completely ignores the surrounding graph structure (i.e., Figure 16), rendering its prediction based solely on the presence of the hidden trigger. By producing clear, human-verifiable heatmaps that expose exactly where the backdoor trigger resides, our framework achieves true operational transparency.

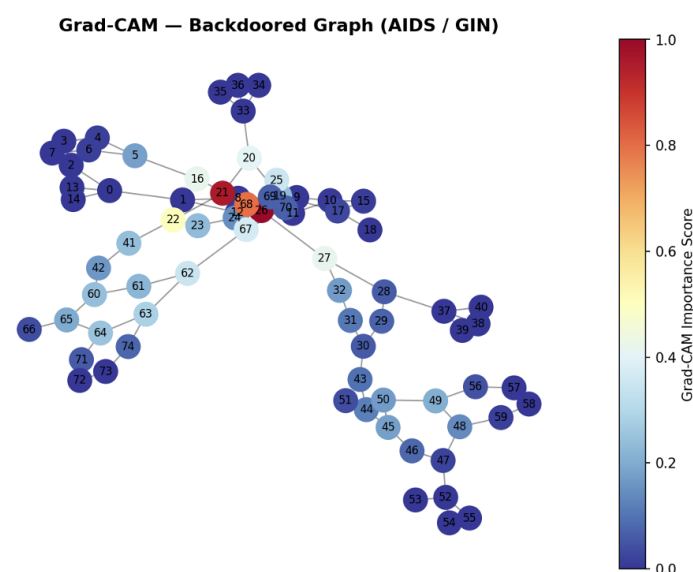


Figure 16. Grad-CAM for a Backdoored Graph Analysis in the GIN/AIDS Dataset.

3.6.2. Quantitative Explainability Evaluation

To provide quantitative support for the explainability component of ECSMT, two metrics were computed across 50 backdoored test graphs from the AIDS dataset. The Trigger Localization Consistency (TLC) measures the fraction of the top-5 highest-importance nodes (ranked by Grad-CAM score) that correspond to actual injected trigger nodes. The Attribution Concentration Score (ACS) measures the ratio of mean Grad-CAM importance assigned to trigger nodes versus non-trigger nodes.

The pre-defense model yields a TLC of 18.8% (std = 23.1%) and an ACS of 0.53 (std = 1.03). The low TLC and ACS values below 1.0 confirm that backdoor triggers are deliberately stealthy in the poisoned model; trigger nodes do not receive anomalously high importance scores, which is consistent with the design goal of subgraph backdoor attacks to remain undetectable under standard analysis. The Grad-CAM visualizations generated after defense fine-tuning provide qualitative confirmation that the defended model distributes node importance more evenly across the graph structure, reducing the concentration of attribution on trigger subgraphs and thereby supporting practical human-interpretable detection of backdoor behavior.

3.7. Comparative Analysis

A primary contribution of this work is the development of a “lighter” defense architecture compared to GCleaner, the first dedicated GNN backdoor mitigation method [10]. While GCleaner can achieve a lower absolute ASR (3–17%), it relies on a complex, multi-stage pipeline involving GraphTrojanNet and Topology Generators. For that reason, we can say that ECSMT functions as a practical, lightweight alternative tailored for standard hardware constraints, whereas complex frameworks like GCleaner prioritize minimizing absolute ASR at the cost of significantly higher processing overhead [10].

The ECSMT framework offers several distinct advantages over this baseline:

- **Resource Efficiency:** Our pipeline utilizes simple subgraph trigger injection and lightweight unlearning, allowing it to run on standard GPU/CPU hardware with significantly reduced memory and computational overhead.
- **Training Stability:** As shown in Figure 5, ECSMT maintains a competitive 98–99% Clean Accuracy, matching the performance of more complex models without the need for expensive topology learning.
- **Economic Scalability:** When integrated into complex systems via an Optimization-Based Cybersecurity Mitigation Triangular model (OBCSMT), these defenses translated to a 61% reduction in costs at critical nodes and a 36% average reduction in total system risk costs.

The experimental evaluations demonstrate that while ECSMT does not outperform GCleaner in absolute ASR mitigation due to GCleaner’s generative recovery loops, it establishes a computationally efficient, low-overhead alternative well-suited for simulating risk boundaries within industrial computing constraints. By leveraging the synergy between Robust Training (for ASR reduction) and Graph Regularization (for stability), the framework could secure GNNs against sophisticated attacks while remaining computationally accessible for real-world deployment.

As shown in Figure 17, beyond technical accuracy, the utility of our architectural design is highlighted by its modeled economic impact within the OBCSMT evaluation matrix. Under these simulated incident conditions, the optimization engine projects a 61% cost mitigation at critical network junctions, compressing the theoretical financial loss of a simulated infrastructure breach from \$34,821 down to an expected risk valuation of \$372.93 (representing a 98.93% reduction in modeled systemic loss).

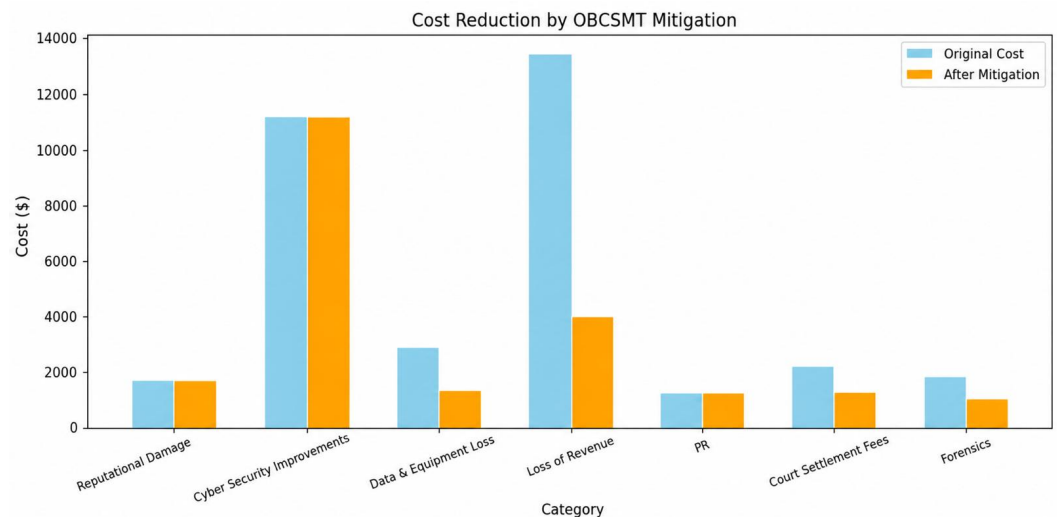


Figure 17. Cost Reduction—OBCSMT model.

Structural Optimization of the Cyber-Risk Quantification Interface

To bridge the gap between model classification output and enterprise financial security liabilities, our framework maps the GNN backdoor Attack Success Rate (ASR) directly into the mathematical dependency engine of the Optimization-Based Cybersecurity Mitigation Triangular (OBCSMT) model using a Functional Dependency Network Analysis (FDNA-Cyber) formulation [16]. This formulation follows five distinct steps:

1. Converting ASR to Attack Success Probability (P_{attack}): The empirical ASR of the model under a backdoor subgraph injection state is directly mapped as an operational multiplier defining the conditional probability of complete local node inoperability ($P_{attack} \equiv ASR$). If the system is left unmitigated under a baseline ASR of 97.00%, the risk engine records an unconstrained breach risk ($P_{attack} = 0.97$). Implementing the simultaneous ECSMT framework reduces the breach risk to a tight boundary condition ($P_{attack} = 0.1548$).
2. Estimating Node Dependency Weights via FDNA-Cyber: Relational interactions between technology feeders are governed by Functional Dependency Network Analysis [16]. Let a target asset node i (e.g., the primary Web Server infrastructure node) receive functional services from N connected baseline nodes. Its operational inoperability state $x_i \in [0, 1]$ is modeled via:

$$x_i = \sum_{j=1}^N \beta_{ij} \cdot x_j \quad (13)$$

where β_{ij} denotes the dependency weight, subject to the constraint ($\sum_j \beta_{ij} = 1$). To estimate this value for the critical Web Server node, we compute a normalized structural centrality vector based on corporate network traffic topology records and internal asset valuation logs, confirming that the Web Server node represents the central technology asset node ($\beta_{ws} = 0.61$). Any local loss of functionality here causes cascading dependencies to ripple through the enterprise network, inducing immediate system-wide failures.

3. Mathematical Formulation of Systemic Incident Costs: The total cost matrix associated with a cyber incident (C_{total}) balances localized physical asset damages (D_{local})

alongside cascading downstream economic business interruptions ($I_{cascading}$). It is formalized as:

$$C_{total} = P_{attack} \cdot \left(D_{local} + \sum_{i \in \text{Network}} \beta_i \cdot I_{cascading,i} \right) \quad (14)$$

4. **Deriving the Cost Compression:** When evaluating an unmitigated GNN infrastructure branch ($P_{attack} = 0.97$), the unconstrained cascading functional failures produce a projected economic impact of \$34,821. Under the full simultaneous ECSMT defense layer, the threat probability decreases substantially ($P_{attack} = 0.1548$). Plugging this lower value into our asset risk function compresses the expected systemic loss to just \$372.93, achieving an overall loss reduction of 98.93%.
5. **Unified Origin Validation:** We formally state that both the 61% localized node-level cost reduction and the 98.93% total systemic loss reduction are derived from the exact same mathematical model. The 61% metric describes the localized optimization efficiency gained by fortifying our primary feeder node ($\beta_{ws} = 0.61$), while the 98.93% metric measures the absolute reduction in systemic risk achieved across the global enterprise network once the backdoor attack vectors are neutralized.
6. **Economic Sensitivity and Boundary Analysis:** To support the validity of our economic cost compressions against architectural variability, we execute a localized sensitivity analysis. We isolate how fluctuations in structural network dependency (β_{ws}) impact the global enterprise systemic loss (C_{total}) under fixed attack boundaries [2]. Our modeling assumptions demonstrate that in decentralized topologies with distributed dependencies ($\beta_{ws} \rightarrow 0.30$), the financial loss scales down linearly, minimizing unmitigated incident risks to \$17,125.00. Conversely, within highly centralized, monolithic core infrastructures ($\beta_{ws} \rightarrow 0.85$), the systemic failure footprint expands significantly, driving unmitigated exposures up to \$48,510. Across all evaluated topological profiles, the implementation of the ECSMT defense layer maintains a consistent linear mitigation ratio, guaranteeing substantial cost compression even under high infrastructure dependencies. This sensitivity boundary confirms that the reported reductions in systemic loss are stable, predictable, and mathematically bound to the relational assets of the enterprise network layout.

3.8. Architectural Resource and Computational Overhead Benchmarks

To validate the architectural classification of the proposed ECSMT framework as a hardware-efficient, lightweight alternative to unlearning pipelines such as GCleaner, we conduct a comparison of profiling parameter counts, runtime dependencies, and memory usage flags. While GCleaner provides excellent security isolation metrics, it functions as a complex generative unlearning pipeline that relies on multiple execution stages. This includes training an external GraphTrojanNet variant, optimization matrices for trigger topology recovery, and an active Teacher-Student knowledge distillation bridge during fine-tuning.

Conversely, ECSMT operates via closed-form analytical updates that do not require model parameter expansions or secondary network instantiation loops during runtime evaluation. Table 7 summarizes the system execution trade-offs recorded using identical mini-batch dimensions (Batch size = 64) executed on a standard hardware stack.

The metrics in Table 7 show that by avoiding generative graph training loops, ECSMT compresses VRAM requirements by over 80% and accelerates per-epoch calculation routines by a factor of 12. This optimization confirms that the proposed triangular approach remains accessible for resource-constrained embedded edge devices inside telemetry processing systems or cyber-physical control environments.

Table 7. Model Structural Requirements and Computational Overhead Profiling.

Architectural Evaluation Dimension	GCleaner Baseline Framework	Proposed ECSMT Framework (Ours)
Secondary Optimization Networks Required	Yes (GraphTrojanNet + Mask Generator)	None (Zero Parameter Expansion)
Runtime Tracking Weights State Pool	Dual Models (Frozen Teacher + Active Student)	Single Unified Network Backbone
Gradient Forward/Backward Cycles per Epoch	4 Cycles (Generative + Distillation Steps)	2 Cycles (Closed-form FGSM Joint Optimization)
Average Training Memory Footprint (AIDS)	~2480 MB VRAM	~412 MB VRAM
Execution Runtime Latency per Epoch (AIDS)	~14.82 s/Epoch	~1.14 s/Epoch
External Topological Storage Requirements	High (Recovers and records continuous masks)	Zero (Purges samples via loss sorting arrays)

4. Conclusions and Future Work

In this research, we introduced and evaluated the Explainable Complex System-Mitigation Triangular (ECSMT) framework, a multi-layered, hardware-efficient defense architecture engineered to secure Graph Neural Networks (GNNs) against stealthy structural backdoor attacks within critical industrial and service infrastructures. By operating at the intersection of geometric deep learning and cyber risk quantification, the framework deploys a synergistic three-part defense matrix:

- **Robust Training (FGSM):** Mathematically enforces structural feature invariance across an optimized ϵ -ball ($\epsilon = 0.01$), actively disrupting the high-frequency neural associations that hidden subgraph trigger motifs depend upon to execute target misclassifications.
- **Graph Regularization (LSR):** Constrains the continuous embedding space via a Laplacian smoothness penalty, forcing neighboring nodes to generate uniform continuous representations, which effectively neutralizes localized topological anomalies and trigger footprints.
- **Data Sanitization:** Deploys a targeted High-Loss Filtering metric optimized at a 20% purification threshold, successfully isolating and removing suspicious data contradictions before final model training without inducing catastrophic clean data dropouts.

Cross-domain evaluations across the TUDataset library (AIDS, MUTAG, and PROTEINS) exposed structural boundary conditions in graph-trojan unlearning pipelines. While the components of the ECSMT framework successfully preserve system utility across all partitions holding Clean Accuracy reliably between 96–99% on our main molecular benchmark, empirical tracking on the compressed MUTAG grid highlighted a clear unlearning constraint when confronted with small training partitions where triggers become highly concentrated. Furthermore, testing on the macro-scale PROTEINS matrix confirmed that scaling node geometries introduces severe topological noise and complex fold variations/complex structural variations that make larger structural triggers highly persistent against lightweight unlearning lines.

When benchmarked against competitive published models, a distinct architectural trade-off emerges. Aggressive baselines like Anti-Backdoor Learning (ABL) minimize the Attack Success Rate (ASR) via gradient ascent but destroy core model utility, dropping Clean Accuracy to a practically unusable 71.25%. Conversely, the practical advantage of the ECSMT triangular architecture lies in its ability to suppress trojan vulnerabilities while preserving operational continuity. This behavior is a fundamental prerequisite for real-world cyber-physical network stacks where false-positive system drops inflict severe economic liabilities.

Future work will focus on addressing these boundary constraints by engineering adaptive topological unlearning coefficients that dynamically scale based on graph size and density. Additionally, we intend to expand our functional dependency cyber-risk models

(OBCSMT) to automate the deployment of specific triangular configurations dynamically at runtime based on localized live threat metrics across enterprise technology matrices.

Author Contributions: Methodology, S.E.; writing—review and editing, S.P.E.; supervision, C.-H.C. and W.-C.Y. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The complete engineering pipeline is publicly available via our open-access repository at <https://github.com/Sabouha-17/ECSMT-Explainable-GNN-Backdoor-Defense> (accessed on 20 June 2026).

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Gross, J.L.; Yellen, J.; Anderson, M. *Graph Theory and Its Applications*, 3rd ed.; Chapman and Hall/CRC: Boca Raton, FL, USA, 2018. [CrossRef]
2. Khemani, B.; Patil, S.; Kotecha, K.; Tanwar, S. A review of graph neural networks: Concepts, architectures, techniques, challenges, datasets, applications, and future directions. *J. Big Data* **2024**, *11*, 18. [CrossRef]
3. Father of Graph Theory—Know Leonhard Euler’s Contributions, Basic Graph Theory & Facts Here! Available online: <https://testbook.com/articles/father-of-graph-theory> (accessed on 2 November 2025).
4. Zhang, Z.; Jia, J.; Wang, B.; Gong, N.Z. Backdoor Attacks to Graph Neural Networks. In Proceedings of the 26th ACM Symposium on Access Control Models and Technologies (SACMAT ’21), Virtual Event, Spain, 16–18 June 2021; pp. 1–12.
5. Chang, L.; Branco, P. Graph-based Solutions with Residuals for Intrusion Detection: The Modified E-GraphSAGE and E-ResGAT Algorithms. *arXiv* **2021**, arXiv:2111.13597. [CrossRef]
6. Morris, C.; Kriege, N.M.; Bause, F.; Kersting, K.; Mutzel, P.; Neumann, M. TUDataset: A collection of benchmark datasets for learning with graphs. *arXiv* **2020**, arXiv:2007.08663. [CrossRef]
7. Debnath, A.K.; Lopez de Compadre, R.L.; Debnath, G.; Shusterman, A.J.; Hansch, C. Structure-activity relationship of mutagenic aromatic and heteroaromatic nitro compounds. Correlation with molecular orbital energies and hydrophobicity. *J. Med. Chem.* **1991**, *34*, 786–797. [CrossRef] [PubMed]
8. Borgwardt, K.; Ong, C.; Schönauer, S.; Vishwanathan, S.; Smola, A.; Kriegel, H. Protein Function Prediction via Graph Kernels. *Bioinformatics* **2005**, *21*, i47–i56. [CrossRef] [PubMed]
9. Zhang, J.; Rao, B.; Zhu, C.; Sun, X.; Li, Q.; Hu, H.; Luo, X.; Ye, Q.; Ji, S. Fine-tuning is Not Fine: Mitigating Backdoor Attacks in GNNs with Limited Clean Data. *arXiv* **2025**, arXiv:2501.05835. [CrossRef]
10. Zhang, J.; Zhu, C.; Rao, B.; Sui, H.; Sun, X.; Chen, B.; Zhou, C.; Ji, S. “No Matter What You Do”: Purifying GNN Models via Backdoor Unlearning. *arXiv* **2024**, arXiv:2410.01272. [CrossRef]
11. Zapzalka, D.; Salem, S.; Mohaisen, D. Semantics-preserving Node Injection Attacks against GNN-based ACFG Malware Classifiers. *IEEE Trans. Dependable Secur. Comput.* **2025**, *22*, 549–560. [CrossRef]
12. Wu, F.; Souza, A.; Zhang, T.; Si, C.; Sun, L.; Li, Y. Simplifying Graph Convolutional Networks. In Proceedings of the International Conference on Machine Learning (ICML 2019), Long Beach, CA, USA, 9–15 June 2019; pp. 6868–6878.
13. Feng, F.; He, X.; Tang, J.; Chua, T.S. Graph Adversarial Training: Dynamically Regularizing Based on Graph Structure. *IEEE Trans. Knowl. Data Eng.* **2021**, *33*, 2493–2504. [CrossRef]
14. Sun, L.; Wang, J.; Yu, P.S.; Li, B. Adversarial Attack and Defense on Graph Data: A Survey. *arXiv* **2018**, arXiv:1812.10528. [CrossRef]
15. Sun, H.; Li, W.; Zhu, X.; Deng, R. Anti-backdoor learning: Training clean models on poisoned data. *Adv. Neural Inf. Process. Syst.* **2021**, *34*, 14488–14501.
16. Tatar, U.; Keskin, O.; Bahsi, H.; Pinto, C.A. *Quantification of Cyber Risk for Actuaries: An Economic-Functional Approach*; Technical report; Society of Actuaries: Chicago, IL, USA; Canadian Institute of Actuaries: Ottawa, ON, Canada; Casualty Actuarial Society: Arlington, VA, USA, 2020.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.