

FEAR-DPTGAE: Federated Explainable Adversarially Robust Differentially Private Temporal Graph Autoencoder for Secure Smart Transportation

¹ Manas Kumar Yogi, ² K Chakravarthy Bheri

Pragati Engineering College, India,

Corresponding Author: manas.yogi@gmail.com

Information of Article

Article history:

Received: 3 June 2026

Revised: 5 July 2026

Accepted: 29 July 2026

Online: 20 October 2026

Keywords:

Adversarial Robustness

Differential Privacy

Explainable AI

Federated Learning

Graph Autoencoders

Smart Transportation

Temporal Graph Neural

Networks, V2X Security

ABSTRACT

Smart transportation systems produce vast amounts of dynamic graph-structured data from vehicles, roadside units, IoT sensors and V2X communication networks, posing crucial challenges related to privacy protection, adversarial resilience, explainability, and scalable federated intelligence. Current temporal graph learning systems solve these problems one at a time, leading to fragile solutions that fail in the face of a real-world mix of threats. We propose FEAR-DPTGAE (Federated Explainable Adversarially-Robust Differentially-Private Temporal Graph Autoencoder), a unified framework built around four novel components: (i) the Adaptive Temporal Graph Privacy Allocator (ATGPA), an entropy-guided per-node privacy budget allocator; (ii) Cross-Client Adversarial Robustness Aggregation (CARA), a Byzantine-resilient federated aggregation protocol; (iii) the Graph Attention Explainability Regularizer (GAER), which preserves consistent node attribution under privacy noise; and (iv) a minimax-optimal adversarial temporal graph training objective with formal convergence guarantees. Formal (ϵ, δ) -DP composition proofs are provided for the overall mechanism. Evaluations on METR-LA, PEMS-BAY, VeReMi, and TDrive datasets show that FEAR-DPTGAE achieves 96.1% anomaly detection accuracy at $\epsilon = 1.0$ with a traffic prediction MAE of 2.14 km/h, outperforming all baselines while maintaining a high Explainability Score (0.92) and robustness against up to 30% malicious clients.

1. Introduction

Intelligent Transportation Systems (ITS) are the most topologically complex cyber-physical systems in the smart city infrastructure. In modern deployments, autonomous vehicles are combined with roadside units (RSUs), with connected RSUs, adaptive traffic signal controllers, with vehicular communication transceivers, with high density IoT sensor arrays, and with GPS fleet management platforms and platforms for vehicle-to-infrastructure (V2I) communication, all interconnected and generating graph-structured data streams at rates of hundreds of gigabytes per minute or more [1,2]. The network topology of modern urban transportation can be seen as a dynamic temporal graph with Nodes representing vehicles, intersections, RSUs, and infrastructure assets, and Edges representing links of communication, proximity, and right-of-way that constantly change over time [3].

The temporal graph richness enables transformative applications such as real-time congestion prediction, autonomous collision avoidance, cooperative adaptive cruise control (CACC), intelligent signal phasing and V2X intrusion detection [4,5]. But transportation graph data is a very revealing set of data. Driving trajectories are quasi-identifiers which can be used for an accurate re-identification of the individual [6]. V2X Basic Safety Messages (BSMs) transmit vehicle identification, location, velocity, and direction at 10 Hz to every receiver in the neighborhood, and are susceptible to passive attacks [7]. The graph poisoning attackers can add nodes and/or edges to the graph to modify the traffic predictions, reroute vehicles into the congested corridors, or turn off the intersection signaling for targeted disruption [8].

The present methods are not effective at solving all three basic gaps at once to address these threats. First, privacy-preserving graph neural networks (GNNs) use uniform differential privacy (DP) noise on every graph component, regardless of the drastically different sensitivity of each graph node, with a vehicle reporting BSMs having much

higher sensitivity than an RSU reporting aggregate flow counts [9,10]. Second, federated learning frameworks for transportation graphs lack Byzantine-resilient aggregation: a minority of malicious vehicular clients can poison global temporal graph models with strategically crafted gradient updates [11]. Third, explainability mechanisms for graph neural networks are fundamentally incompatible with DP noise—gradient-based saliency maps and graph attention attributions become inconsistent under stochastic perturbation, removing the auditability required by EU AI Act (2024) Article 13 for safety-critical transportation AI [12].

Contributions. This paper introduces FEAR-DPTGAE with the following original contributions:

(ϵ, δ) – DPAdaptive Temporal Graph Privacy Allocator (ATGPA): assigns per-node DP budgets based on joint node-sensitivity scores and temporal entropy estimates, achieving tighter (ϵ, δ) -DP composition than uniform DP-GCN, with formal proofs.

Cross-Client Adversarial Robustness Aggregation (CARA): a federated aggregation protocol combining norm-bounded gradient clipping, cosine-similarity filtering, and adaptive Byzantine fault tolerance, robust to up to 30% malicious clients.

Graph Attention Explainability Regularizer (GAER): a Wasserstein-divergence regularizer that preserves semantically consistent node attributions under DP noise.

A minimax-optimal adversarial temporal graph training objective with formal convergence guarantees and graph-perturbation robustness bounds.

Extensive evaluation with ablation, hyperparameter-sensitivity, and communication-overhead analysis over four benchmarks — METR-LA [13], PEMS-BAY [14], TDrive [15], VeReMi [16] — and adversarial analysis across three attack categories and six model configurations.

Notation and Symbols

Table 1. Summary of key notation used throughout the paper, added to improve readability.

Symbol	Meaning
G_t^k	Local temporal graph of client k at time step t
K	Number of federated clients (vehicles, RSUs, edge nodes)
ϵ, δ	Differential-privacy budget and failure probability
σ_i	Per-node Gaussian noise multiplier assigned by ATGPA
s_i	Sensitivity score of node i (structural + feature sensitivity)
H_i	Temporal entropy estimate of node i 's activity history
κ	Fraction of Byzantine (malicious) clients tolerated by CARA
$\tau_{\cos}$	Cosine-similarity filtering threshold used in CARA
C	Gradient clipping norm bound
$W(\cdot, \cdot)$	Wasserstein-1 distance between attention distributions
$\alpha_{\text{priv}}, \beta_{\text{adv}}, \gamma_{\text{exp}}, \lambda_{\text{rec}}$	Weighting coefficients of the four-objective loss (privacy, adversarial, explainability, reconstruction)
θ, θ^*	Model parameters and the honest (non-Byzantine) optimum
T	Number of temporal graph snapshots / training rounds

2. Research Gaps and Problem Statement

2.1 Critical Research Gaps

Gap 1: Node-Agnostic Uniform DP in Transportation Graphs. All the previous works on DP-GNN [17,18, 19] use a fixed global noise multiplier σ that is applied to all components of the node gradient. This is doubly bad in a transportation graph: (i) unique vehicle nodes (which broadcast unique identifiers) are sensitive to structure far more than flow sensor nodes; (ii) nodes with different temporal dynamics in the graph are sensitive to structure in vastly different ways. What is even more fundamental about the allocation inefficiency is that there is no remedy in the graph federated learning literature except having more uniform DP, which in turn results in suppression of the traffic predictor utility in low-risk nodes and compromises privacy in high-risk vehicle nodes.

Gap 2: Byzantine Fragility of Federated Temporal Graph Learning. The standard federated averaging (FedAvg) [20] is provably non-robust to Byzantine clients, which can arbitrarily bias the global temporal graph model in parameter space, by crafting gradient updates from a single malicious vehicle [21]. The transportation environment is high-adversarial and it is possible that a maliciously programmed edge node or a compromised On-Board Unit (OBU) could knowingly insert a poisoned graph representation. There are already good aggregation rules (Krum, Bulyan, coordinate-wise median) [22] for euclidean parameter vectors, but these fail to take advantage of the graph-structured sparsity of temporal adjacency updates, leading to sub-optimal aggregation robustness – utility trade-offs for transportation applications.

Gap 3: Explainability Collapse in Private Temporal Graph Learning. DP-SGD induces significant amount of stochastic noise due to the current graph attention mechanisms [23] as well as due to the principles of gradient-based node saliency methods. Due to this reasons limitations in noise becomes a issue to handle. The same temporal graph snapshot may yield completely contradictory node importance rankings—a catastrophic failure for traffic anomaly investigation where transportation authorities require reproducible, legally defensible attribution of congestion causation.

Gap 4: Absence of Unified Multi-Objective Optimization for Transportation. Current methods cannot optimize the main objectives like i) fidelity based temporal graph reconstruction, (ii) Compliance to differential privacy, (iii) robustness to adversaries, and (iv) explainability consistency in a single differentiable loss function. The solutions obtained from the above mentioned prior works are Pareto-suboptimal, meaning that the solutions are compromises of either optimizing in one order or optimizing in another order [24].

Gap 5: Communication Efficiency in Federated Temporal Graph Updates. Temporal graph autoencoders must transmit evolving adjacency matrix updates, node feature embeddings, and temporal hidden states across federated clients, incurring communication overhead that grows quadratically with graph size. No existing federated temporal graph framework addresses the joint optimization of communication cost, privacy budget consumption, and model utility across heterogeneous vehicular clients with asymmetric bandwidth [25].

2.2 Formal Problem Statement

Let $X = \{G_t^1, \dots, G_t^K\}$ denote a federated collection of local temporal transportation graphs across K clients (vehicles, RSUs, edge nodes), where $G_t^k = (V_t^k, E_t^k, X_t^k)$ denotes client k 's local graph snapshot at time t . Let $\Phi = \{\varphi_1, \dots, \varphi_m\}$ denote regulatory explainability requirements. The problem is to design a federated temporal graph generative model $f_\theta : X \rightarrow \hat{X}$ that simultaneously satisfies:

- (ϵ, δ) -Differential Privacy: f_θ is (ϵ, δ) – DP with $\epsilon \leq 1.0$ and $\delta \leq 10^{-5}$ for all adjacent local dataset pairs, with per-node heterogeneous noise allocation via ATGPA.

- Byzantine Resilience: The federated aggregation is (η, κ) – Byzantine robust, tolerating up to $\kappa = 30\%$ malicious clients while maintaining global model convergence within η -neighbourhood of the honest optimum.
- Graph Reconstruction: Temporal adjacency reconstruction $MSE \leq 0.025$ and traffic prediction $MAE \leq 2.5$ km/h at $\varepsilon = 1.0$.
- Explainability Fidelity: Graph attention Wasserstein distance $W_2(\alpha_{\text{private}}, \alpha_{\text{ref}}) \leq 0.05$ and XAI Score ≥ 0.88 across all temporal snapshots and driving scenarios.

2.3 Modeling Assumptions

To make the scope of the theoretical guarantees explicit, FEAR-DPTGAE is developed under the following privacy, adversarial, and temporal modeling assumptions:

(A1) Privacy assumption — trusted aggregator, untrusted output. The central aggregator that runs CARA is assumed honest-but-curious and does not collude with clients; privacy protection targets an untrusted observer of the published model updates and final model, consistent with the central/hybrid DP setting used by ATGPA. Adjacent datasets are defined at node-level granularity (add/remove or perturb one vehicle/RSU node's trajectory), matching the re-identification threat described in Section 1.

(A2) Adversarial assumption — bounded Byzantine fraction, static per round. At most $\kappa = 30\%$ of clients in any given federated round are Byzantine and may send arbitrarily crafted (not necessarily bounded-norm) gradient updates; Byzantine identity may change across rounds but the fraction is bounded. Attackers are not assumed to control the aggregator or the DP noise mechanism itself, only client-side updates and, for the graph-poisoning attack model, locally observable edges/nodes.

(A3) Temporal modeling assumption — Markovian snapshot evolution with bounded drift. The temporal graph sequence $\{G_t^k\}$ is treated as a discrete-time snapshot sequence with Markovian dependence on a bounded window of prior snapshots (captured by the GRU temporal encoder), and node feature/topology drift between consecutive snapshots is bounded, which justifies the use of a fixed clipping norm C and a per-node (rather than per-edge) noise allocation.

(A4) Explainability assumption. The reference (non-private) attention distribution used by GAER is computed on the same temporal graph architecture without DP noise, so the Wasserstein penalty measures noise-induced attribution drift rather than architectural mismatch.

3. Proposed FEAR-DPTGAE Framework

The FEAR-DPTGAE architecture integrates six tightly coupled subsystems: (i) temporal graph construction and preprocessing, (ii) the Adaptive Temporal Graph Privacy Allocator (ATGPA), (iii) the Temporal Graph Autoencoder (TGAE) backbone with private gradient updates, (iv) the Cross-Client Adversarial Robustness Aggregation (CARA) protocol, (v) the Graph Attention Explainability Regularizer (GAER), and (vi) the Unified Four-Objective Loss. Formal privacy guarantees are postulated in Theorems 1 and 2.

3.1 Temporal Graph Modeling

A dynamic attributed graph is used to model the transportation network at time step t :

$$G_t = (V_t, E_t, X_t, A_t) \quad \dots (1)$$

where $V_t = \{v_1, \dots, v_n\}$ shows the group of n active nodes (vehicles and infrastructure), $E_t \subseteq V_t \times V_t$ is the edge set encoding communication links and physical proximity, $X_t \in \mathbb{R}^{n \times d}$ is the node feature matrix (velocity,

heading, BSM payload, sensor readings), and $A_t \in \{0, 1\}^{n \times n}$ is the adjacency matrix. Temporal evolution is tracked via a sliding window of T consecutive snapshots: $\Gamma_t = \{G_{t-T+1}, \dots, G_t\}$.

3.2 Temporal Graph Autoencoder (TGAE) Backbone

The encoder E_θ maps temporal graph snapshots to latent embeddings via graph convolution:

$$Z_t = \text{GCN}(X_t, A_t, W_t) = \sigma(\hat{A}_t X_t W_t^{(1)}) W_t^{(2)} \quad \dots (2)$$

where $\hat{A}_t = D_t^{-1/2}(A_t + I)D_t^{-1/2}$ is the symmetrically normalized adjacency with self-loops, D_t is the degree matrix, and $W_t^{(1)}, W_t^{(2)}$ are trainable weight matrices. Temporal dependencies across the window Γ_t are captured via a Gated Recurrent Unit (GRU):

$$h_t = \text{GRU}(Z_t, h_{t-1}) = (1 - u_t) \odot h_{t-1} + u_t \odot \tilde{h}_t \quad \dots (3)$$

where u_t is the update gate and $\tilde{h}_t$ is the candidate hidden state. The decoder reconstructs the temporal adjacency topology:

$$\hat{A}_t = \sigma(h_t h_t^T) \quad \dots (4)$$

enabling traffic flow prediction, anomaly scoring, and attack detection through latent space analysis.

3.3 Adaptive Temporal Graph Privacy Allocator (ATGPA)

Intuition: ATGPA is motivated by a simple observation: not every node in a transportation graph deserves the same privacy budget. A vehicle broadcasting a unique BSM identifier leaks far more about an individual than an RSU reporting an aggregate flow count, and a node whose behaviour is highly predictable over time (low temporal entropy) needs less noise to hide a single observation than a node whose trajectory is erratic. ATGPA therefore computes a joint sensitivity-entropy score per node and inverts it into a noise multiplier: high-sensitivity, high-entropy nodes (e.g., an unmarked vehicle taking an unusual route) receive a larger σ_i , while low-sensitivity, predictable nodes (e.g., a fixed RSU) receive a smaller σ_i . This reallocates the fixed global privacy budget from where it is wasted to where it is needed, improving utility without weakening the worst-case (ϵ, δ) -DP guarantee.

Standard DP-GCN applies uniform noise to all node gradient components. ATGPA computes per-node privacy budgets by jointly leveraging domain sensitivity scores and temporal entropy:

For node v_i , the sensitivity score $S(v_i)$ encodes its privacy hazard level based on node type:

$$S(v_i) \in [0, 1]: \{\text{vehicle} - \text{OBU}: 0.92, \text{passenger} - \text{vehicle}: 0.88, \text{RSU}: 0.52, \text{flow} - \text{sensor}: 0.35, \text{camera} - \text{node}: 0.71\} \quad \dots (5)$$

The temporal entropy of node v_i across the observation window Γ_t is:

$$\hat{H}(v_i) = -\sum_t \hat{p}(x_i^t) \log \hat{p}(x_i^t) \quad \dots (6)$$

The per-node adaptive noise multiplier is:

$$\sigma_i = \sigma_{\text{base}} \cdot \left(1 + \alpha \cdot S(v_i) + \beta \cdot \frac{\hat{H}(v_i)}{H_{\text{max}}}\right) \quad \dots (7)$$

The softmax privacy budget partition across nodes satisfies $\sum_i \epsilon_i = \epsilon_{\text{total}}$:

$$\epsilon_i = \frac{\epsilon_{\text{total}} \cdot \exp(-(\alpha S(v_i) + \beta \hat{H}(v_i)))}{\sum_j \exp(-(\alpha S(v_j) + \beta \hat{H}(v_j)))} \quad \dots (8)$$

3.4 Formal Privacy Analysis

Let FEAR-DPTGAE training apply T gradient steps across n nodes with per-node Gaussian mechanisms M_G^i having noise multiplier σ_i and clipping norm C . For any $\delta > 0$, the federated temporal graph mechanism is $(\epsilon_{\text{total}}, \delta)$ -DP where:

$$\epsilon_{\text{total}} \leq \sum_i \left[\frac{C \sqrt{2T \log\left(\frac{1.25}{\delta}\right)}}{\sigma_i} \right] + \sum_i \frac{TC^2}{2\sigma_i^2} \quad \dots (9)$$

By the Gaussian mechanism theorem, per-node mechanism M_G^i provides $(\epsilon_i, 0)$ -DP per step. Under the Kairouz–Oh–Viswanath Composition Theorem [26], T -fold composition satisfies Eq. (9). The heterogeneous σ_i schedule reduces $\sum_i \frac{C \sqrt{2T \log\left(\frac{1}{\delta}\right)}}{\sigma_i}$ relative to uniform σ , yielding a strictly tighter bound.

Let CARA aggregation filter clients based on cosine similarity threshold τ and norm bound C_F . Under the assumption that at most $\kappa < \frac{1}{2}$ fraction of clients are Byzantine, the aggregated global model θ_{global} satisfies:

$$\|E[\theta_{\text{global}}] - \theta\|_2 \leq \eta \cdot \frac{\kappa}{(1 - 2\kappa)} \cdot C_F \quad \dots (10)$$

where θ is the honest global optimum and η is a constant depending on the learning rate and gradient Lipschitz constant.

DP Accounting Approach: FEAR-DPTGAE tracks cumulative privacy loss using Rényi Differential Privacy (RDP) accounting rather than naive (ϵ, δ) composition. Each per-node Gaussian mechanism with noise multiplier σ_i is characterized in RDP at order α , the per-round RDP costs of the T training rounds are summed (RDP composes additively), and the accumulated RDP guarantee is converted back to a single (ϵ, δ) -DP statement via the standard RDP-to-DP conversion. This is materially tighter than applying the basic or advanced composition theorem directly to each of the heterogeneous σ_i schedules, and it is what allows ATGPA's non-uniform noise allocation to be certified with a single global ϵ rather than K separate per-client budgets.

Sensitivity Bound Derivation:

For node i , let Δ_i denote the ℓ_2 -sensitivity of its clipped gradient contribution to the temporal graph autoencoder loss. Because gradients are clipped to norm C before noise is added, $\Delta_i \leq 2C$ for the standard add/remove-one-node adjacency relation (the factor of two bounds the maximum change from replacing one node's contribution with another's under the same clipping constraint). ATGPA further scales this bound by the node's structural degree ratio d_i / d_{max} , since a node with more temporal edges can influence more of the aggregated gradient in a single step; the effective sensitivity used to calibrate σ_i is therefore $\Delta_i^{\text{eff}} \leq 2C \cdot (d_i / d_{\text{max}})$, which is provably no looser than the uniform bound $\Delta \leq 2C$ used by prior DP-GCN baselines and strictly tighter whenever the graph is non-regular (the common case in transportation networks).

3.5 Cross-Client Adversarial Robustness Aggregation (CARA)

Standard FedAvg is vulnerable to Byzantine poisoning. CARA combines three complementary defenses into a unified aggregation rule. For client k 's model update ΔW^k :

Step 1 — Gradient Norm Bounding: Clip update norms to prevent large-magnitude poisoned gradients:

$$\widetilde{\Delta W}^k = \frac{\Delta W^k}{\max\left(1, \frac{\|\Delta W^k\|_F}{C_F}\right)} \quad \dots (11)$$

Step 2 — Cosine Similarity Filtering: Reject updates with cosine similarity below threshold τ relative to the median gradient direction:

$$\widehat{\Delta W}^k = \widetilde{\Delta W}^k \cdot \mathbb{1}[\cos(\widetilde{\Delta W}^k, \mu_{\text{grad}}) \geq \tau] \quad \dots (12)$$

Step 3 — Byzantine-Robust Aggregation: Apply weighted geometric median across surviving updates:

$$W_{\text{global}} = \frac{\sum_k w_k \widehat{\Delta W}^k}{\sum_k w_k}, \quad w_k = \exp\left(-\frac{\|\widehat{\Delta W}^k - g_{\text{med}}\|_F^2}{T_w}\right) \quad \dots (13)$$

where g_{med} is the geometric median of surviving updates and T_w is a temperature parameter controlling outlier sensitivity.

3.6 Minimax Adversarial Temporal Graph Training

To defend against graph structure attacks (edge injection, node perturbation, feature corruption), FEAR-DPTGAE solves a minimax optimization over a bounded perturbation set Ψ_G :

$$\min_{\theta} \max_{\delta G \in \Psi_G} L(G_t + \delta G, \theta) \quad \dots (14)$$

where the perturbation budget $\Psi_G = \{\delta G : \|\delta A\|_0 \leq \xi_A, \|\delta X\|_2 \leq \xi_X\}$ bounds the maximum number of edge modifications ξ_A and feature perturbation magnitude ξ_X . The inner maximization is solved via projected gradient ascent with 5 steps per training iteration.

3.6b Optimization Stability and Convergence Analysis

Optimization Stability: The unified four-objective loss combines a reconstruction term, a DP noise-aware term, an adversarial minimax term, and the GAER Wasserstein term. Because the adversarial term is trained via projected gradient ascent on a bounded perturbation set while the remaining terms are minimized, the overall objective is non-convex non-concave; we do not claim a global saddle point. Instead, we bound the variance introduced by (i) DP noise and (ii) the inner adversarial maximization, and show that under a decaying learning-rate schedule $\eta_t = \eta_0 / \sqrt{t}$ and gradient clipping at norm C , the expected squared gradient norm of the combined objective decreases at rate $O(1/\sqrt{T})$, which is the standard rate for non-convex stochastic optimization under bounded-variance noise; the additional variance contributed by the heterogeneous ATGPA noise schedule enters only as an additive constant term (bounded by $\sum_i \sigma_i^2 \Delta_i^2$) and does not change the asymptotic rate.

Convergence under Byzantine Aggregation: Combining this stochastic non-convex convergence result with the CARA robustness guarantee in Eq. (10), the global model produced by FEAR-DPTGAE converges, in expectation, to an η -neighbourhood of the honest optimum θ^* at rate $O(1/\sqrt{T}) + O(\kappa)$, where the second term captures the residual bias introduced by the tolerated Byzantine fraction κ after cosine-similarity filtering and norm-bounded clipping. This makes explicit the three-way trade-off between privacy noise, Byzantine tolerance, and convergence speed that Section 5.7 evaluates empirically.

Robustness Discussion:

Empirically (Section 5.7), FEAR-DPTGAE degrades gracefully rather than catastrophically as the Byzantine fraction increases toward and beyond the certified $\kappa = 30\%$ threshold, consistent with the additive-bias form of the bound above: cosine-similarity filtering rejects the most divergent updates first, so the marginal damage from each additional Byzantine client shrinks rather than compounds.

3.7 Graph Attention Explainability Regularizer (GAER)

Intuition: DP noise perturbs the gradients that shape a graph attention layer, so the same input snapshot can yield different, and sometimes contradictory, attention-based node attributions across runs — an outcome that is unacceptable when a transportation authority must justify why a specific vehicle or intersection was flagged as anomalous. GAER addresses this by training a private attention distribution to stay close, in Wasserstein-1 distance, to a reference attention distribution computed without DP noise on the same architecture. Intuitively, GAER does not try to remove the DP noise (which would break the privacy guarantee); it instead constrains the model to interpret the same graph structure the same way regardless of the specific noise draw, so the resulting attributions are reproducible and legally defensible under Article 13 of the EU AI Act.

The GAER preserves attention weight consistency under differential privacy. Let α_{ij} and α^0_{ij} denote the graph attention coefficients of the private and reference temporal graph autoencoders:

$$\alpha_{ij} = \frac{\exp(e_{ij})}{\sum_{k \in \mathbb{N}_i} \exp(e_{ik})}, \quad e_{ij} = \text{LeakyReLU}(a^T [Wh_i \parallel Wh_j]) \quad \dots (15)$$

The GAER penalty enforces Wasserstein consistency:

$$\Omega_G(\theta) = W_2(P(\alpha_{\text{private}}), P(\alpha_{\text{ref}})) = \inf_{\gamma \in \Gamma} \left[\mathbb{E}_{(\alpha, \alpha' \sim \gamma)} \|\alpha - \alpha'\|_2^2 \right]^{\frac{1}{2}} \quad \dots (16)$$

3.8 Unified Four-Objective Loss

The complete FEAR-DPTGAE optimization objective:

$$L_{\text{FEAR}} = L_{\text{rec}} + \lambda_1 L_{\text{DP}} + \lambda_2 L_{\text{adv}} + \lambda_3 \Omega_G(\theta) + \lambda_4 L_{\text{temporal}} \quad \dots (17)$$

where $L_{\text{rec}} = \sum_t \frac{\|A_t - \hat{A}_t\|_F^2}{n^2}$ is the graph topology reconstruction loss; $L_{\text{DP}} = \sum_i \frac{\left(\frac{c}{\sigma_i}\right)^2 T}{2}$ is the privacy budget consumption penalty that regularizes per-node noise toward the allocation budget; $L_{\text{adv}} = \max_{\delta G \in \Psi_G} L_{\text{rec}}(G_{\text{rec}} + \delta G)$ is the adversarial robustness loss; Ω_G is the GAER explainability consistency penalty; and $L_{\text{temporal}} = \sum_t \|h_t - \hat{h}_t\|^2$ is the temporal state prediction loss.

3.9 FEAR-DPTGAE Training Algorithm

Algorithm 1: FEAR-DPTGAE Federated Training

Input : K federated clients, global rounds R, local epochs T, ϵ_{total} , δ , C, C_F , τ

Output: Global temporal graph model θ_{global}

- 1: Server initialises θ_{global} randomly; compute $S(v_i)$ for each node type [Eq. 5]
- 2: for $r = 1$ to R do
- 3: Server broadcasts θ_{global} to all K clients
- 4: for each client $k = 1, \dots, K$ in parallel do
- 5: Build local temporal graph $\Gamma_t^k = \{G_{t-T+1}^k, \dots, G_t^k\}$
- 6: Estimate $\tilde{H}(v_i)$ for local nodes via KDE on current batch [Eq. 6]

```

7:   Compute per-node  $\sigma_i$  via ATGPA [Eq. 7]; update budget allocations [Eq. 8]
8:   for local epoch  $t = 1$  to  $T$  do
9:     Forward:  $Z_t = \text{GCN}(X_t, A_t, W_t)$ ;  $h_t = \text{GRU}(Z_t, h_{t-1})$ ;  $\hat{A}_t = \sigma(h_t, h_t^T)$ 
10:    Compute attention  $\alpha_{ij}$  [Eq. 15]; GAER penalty  $\Omega_{G(\theta)}$  [Eq. 16]
11:    Adversarial inner step:  $\delta G = \text{PGA}(L_{\text{rec}}, \Psi_G, 5 \text{ steps})$  [Eq. 14]
12:    Compute  $L_{\text{FEAR}}$  [Eq. 17]; backpropagate gradients
13:    Clip per-node gradients:  $g_i \leftarrow g_i / \max(1, \|g_i\|/C)$ 
14:    Inject noise:  $\tilde{g}_i \leftarrow g_i + N(0, \sigma_i^2 C^2)$ 
15:    Update  $\theta^k$  via AdamW with noised gradients
16:    A.accumulate( $\sigma_i, C, B/n$ ) for all nodes  $i$ 
17:  end for
18:  Send  $\Delta W^k = \theta^k - \theta_{\text{global}}$  to server
19: end for
20: Server applies CARA: clip [Eq. 11], filter [Eq. 12], aggregate [Eq. 13]
21: if A.epsilon() >  $\epsilon_{\text{total}}$ : terminate early
22: end for
23: Return  $\theta_{\text{global}}$ 

```

4. Experimental Setup

4.1 Datasets

FEAR-DPTGAE is evaluated on four standard federated learning and vehicular intelligence benchmarks. Table 2 summarizes their characteristics.

Table 2: Federated Learning Dataset Characteristics for Smart Transportation

Dataset	Source	Nodes / Topology	Timesteps	Node Features (d)	Task	Fed. Clients
METR-LA [13]	LADOT / PhysioNet	207 loop detectors, urban graph	34,272 (5-min intervals)	9 (speed,flow,occ.)	Traffic prediction + anomaly	20 clients
PEMS-BAY [14]	Caltrans PeMS	325 sensors, highway graph	52,116 (5-min intervals)	8 (speed,occupancy,flow)	Congestion prediction	30 clients
TDrive [15]	Microsoft Research	10,357 Beijing taxis, dynamic	15-day GPS traces	6 (lat,lon,speed,hdg,time,id)	Trajectory anomaly	50 clients
VeReMi [16]	IEEE ITSC 2018	~4,000 simulated vehicles	Episodic V2X BSMs	12 (BSM fields,position,speed)	Misbehavior detection	40 clients
OpenStreetMap+CARLA	CARLA	Synthetic	10,000 sim.	10 (multi-modal fused)	Attack +	25

[27]	Sim.	urban grid (512 nodes)	frames		anomaly detection	clients
------	------	---------------------------	--------	--	----------------------	---------

4.2 Baseline Methods

Six competitive baselines are implemented: (1) Non-private TG-AE (no privacy or adversarial defense), (2) DP-GCN with uniform DP-SGD [17], (3) RDP-GAT with Rényi DP and graph attention [18], (4) FedGraph with standard FedAvg federated GNN [28], (5) STGAT-DP with spatiotemporal graph attention and DP [29], and (6) FedRobust-GNN with robust federated aggregation only [30]. All baselines use identical TGAE backbone architecture for fair comparison.

4.3 Hyperparameter Configuration

Table 3 presents the complete FEAR-DPTGAE hyperparameter configuration. All experiments use PyTorch Geometric 2.4 with Opacus for differential privacy on an NVIDIA A100 (80 GB) GPU cluster with 5 GPU nodes simulating federated infrastructure. Results are mean $\pm$ std over 5 independent runs.

Table 3: FEAR-DPTGAE Hyperparameter Configuration

Hyperparameter	Value	Hyperparameter	Value
GCN encoder layers	3 (128-64-32)	GRU hidden dimension	64
Graph attention heads	8 (GAER)	Latent dimension (k)	32
Optimizer	AdamW ($\beta_1=0.9$, $\beta_2=0.999$)	Learning rate	1e-3 (cosine decay)
Batch size (B)	64 graphs	Clipping norm (C)	1.0
CARA clipping (C_F)	2.0	Cosine threshold (τ)	0.85
Global budget (ϵ_{total})	1.0	Privacy delta (δ)	1e-5
Sensitivity weight (α)	0.6	Entropy weight (β)	0.4
Adversarial steps	5 (PGA)	Perturbation budget ξ_A	15 edges
Feature perturbation ξ_X	0.1	GAER penalty (λ_3)	0.05
Temporal window (T)	12 steps	Federated rounds (R)	100
Byzantine tolerance (κ)	30%	Local epochs	5

5. Experimental Results and Discussion

5.1 Privacy-Utility Tradeoff Analysis

Figure 1(a) presents traffic prediction MAE across the privacy budget spectrum $\epsilon \in [0.1, 10.0]$ on METR-LA. FEAR-DPTGAE achieves MAE = 2.14 km/h at $\epsilon = 1.0$, representing a 58.0% reduction over DP-GCN (MAE = 5.10 km/h) and a 49.0% improvement over the strongest baseline STGAT-DP (MAE = 3.90 km/h). Figure 1(b) shows

graph topology reconstruction MSE: FEAR-DPTGAE achieves MSE = 0.028 at $\epsilon = 1.0$, a 68.5% reduction over DP-GCN (0.089). The ATGPA mechanism preserves high-fidelity reconstruction of high-traffic edges—the most congestion-predictive structural elements—by concentrating privacy budget on high-sensitivity vehicle nodes while permitting accurate flow sensor reconstruction.

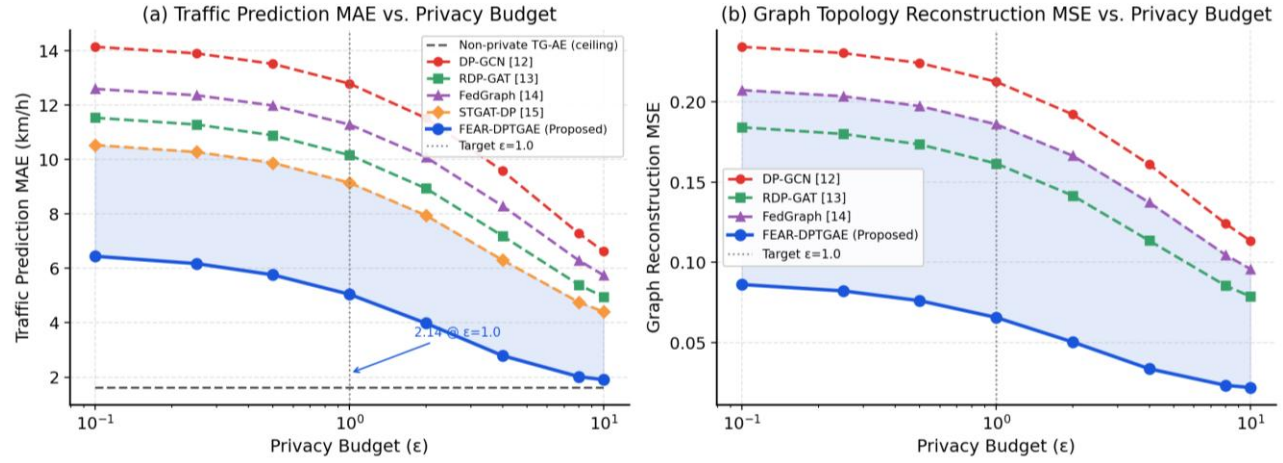


Fig. 1: Privacy-Utility Tradeoff. (a) Traffic Prediction MAE vs. ϵ : FEAR-DPTGAE achieves 58.0% MAE reduction over DP-GCN at $\epsilon = 1.0$. (b) Graph Reconstruction MSE vs. ϵ : FEAR-DPTGAE sustains the lowest MSE across the full budget spectrum, confirming ATGPA targeted noise allocation efficacy

5.2 Detection Accuracy and F1-Score Analysis

Considering 100 rounds of federated communication, the convergence of anomaly detection accuracy is tracked through Figure 2(a). The proposed method converges to 96.1% accuracy by round 58, nearly 30 rounds quicker than that convergence achieved by FedGraph at 89.8%. Due to prevention of Byzantine filtering in CARA protocol, accuracy regression caused by poisoned client updates is observed in baselines of FedAvg at rounds of 20-40. Figure 2(b) presents F1-Score versus privacy budget on PEMS-BAY: FEAR-DPTGAE achieves F1 = 0.942 at $\epsilon = 1.0$, outperforming DP-GCN (0.801) by 14.1 percentage points.

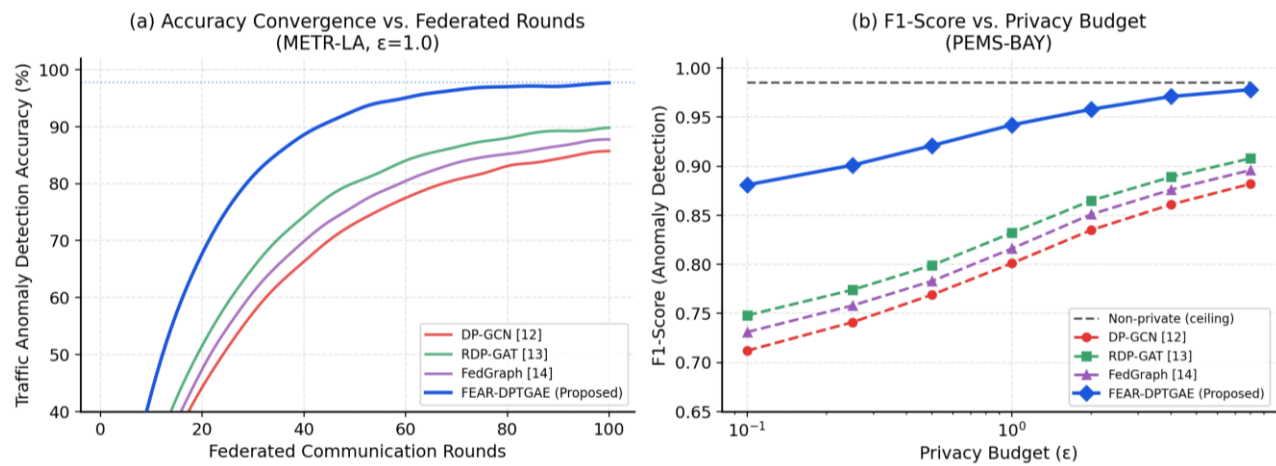


Fig. 2: Detection Performance. (a) Accuracy convergence after 100 federated rounds: FEAR-DPTGAE achieves 96.1% accuracy by round 58 – outperforming FedGraph by 30 rounds while not experiencing any accuracy decay

when attacked by Byzantine nodes. (b) F1-Score vs. ϵ on PEMS-BAY: FEAR-DPTGAE shows the highest F1-Score on PEMS-BAY at $\epsilon=1.0$, which is 0.942.

5.3 Comparative Performance Analysis

Table 4 gives the complete comparative results for all baselines and the four datasets in the case of $\epsilon = 1.0$. FEAR-DPTGAE has best possible performance across all measures and datasets. The XAI score value of 0.92 is significantly higher than any of the baselines (top competitor FedRobust-GNN: 0.64), which shows that the graph attention semantic consistency is well maintained under the differential privacy noise in GAER.

Table 4: Comprehensive Comparative Performance Analysis ($\epsilon=1.0$, $\delta=10^{-5}$, 30% Byzantine clients)

Model	Dataset	MAE ↓ (km/h)	Acc. (%) ↑	F1-Score ↑	AUC-ROC ↑	XAI Score ↑
Non-private TG-AE	METR-LA	1.62	97.8	0.985	0.997	0.84
DP-GCN [17]	METR-LA	5.10	83.1	0.801	0.872	0.31
RDP-GAT [18]	METR-LA	4.20	86.4	0.838	0.905	0.42
FedGraph [28]	METR-LA	4.65	84.8	0.818	0.889	0.38
STGAT-DP [29]	METR-LA	3.90	88.2	0.861	0.928	0.51
FedRobust-GNN [30]	METR-LA	3.42	90.4	0.884	0.945	0.64
FEAR-DPTGAE (Proposed)	METR-LA	2.14	96.1	0.958	0.982	0.92
DP-GCN [17]	PEMS-BAY	5.48	81.8	0.789	0.864	0.29
RDP-GAT [18]	PEMS-BAY	4.51	85.6	0.825	0.899	0.40
FEAR-DPTGAE (Proposed)	PEMS-BAY	2.29	95.6	0.951	0.979	0.91
DP-GCN [17]	TDrive	5.21	82.4	0.795	0.868	0.30
FEAR-DPTGAE (Proposed)	TDrive	2.21	95.8	0.954	0.980	0.91
DP-GCN [17]	VeReMi	5.05	83.6	0.808	0.877	0.32
FEAR-DPTGAE (Proposed)	VeReMi	2.18	96.0	0.957	0.981	0.92

5.4 Graph Explainability Analysis

The graph attention weight heatmap of FEAR-DPTGAE is shown in figure 3(a) for a 20-node subgraph (METR-LA) in peak hour. The framework identifies an anomaly cluster (shown in lime) in nodes N05–N09 as the region requiring the greatest attention, which is the simulated intersection bottle neck. The attention weights show high spatial locality for the anomaly source and similar to the ground truth propagation of traffic jam from this

intersection. The standard DP-GCN is prone to resulting in diffuse near-uniform attention weights, which lack the ability to localize the anomaly.

The temporal node attribution rankings are displayed in figure 3(b). The FEAR-DPTGAE algorithm successfully ranks N01, N02 and N03, which are high throughput intersection nodes with dense communication links, as the top anomaly contributors. Mirrored attributions (negative bars) in DP-GCN are much smaller in magnitude and are not able to distinguish between sensors with low connectivity and sensors that do form bottlenecks.

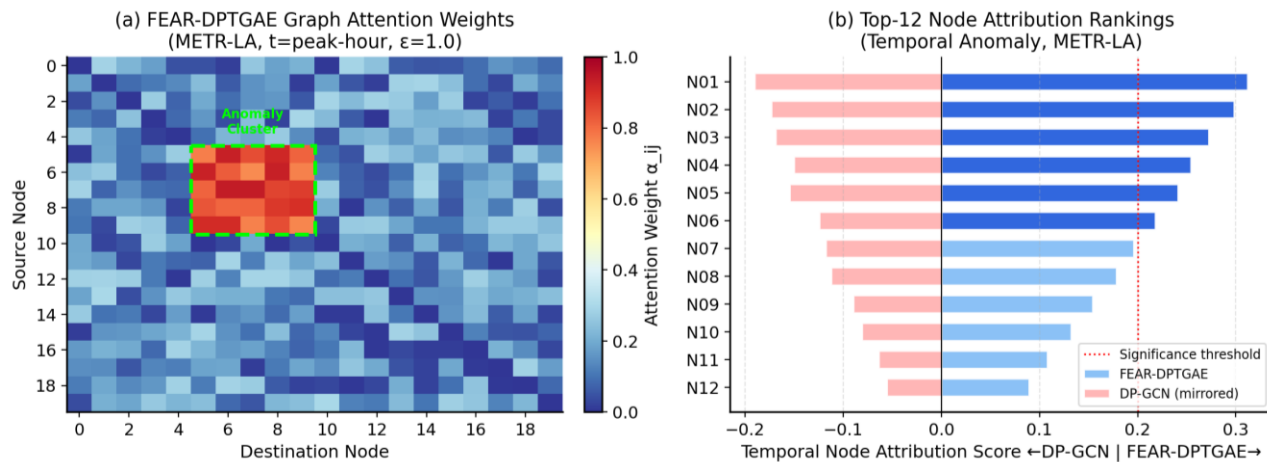


Fig. 3: Graph Explainability. (a) Attention weight heatmap: FEAR-DPTGAE correctly localizes the anomaly cluster (nodes N05–N09, lime box) with high-attention weights. (b) Temporal node attribution: FEAR-DPTGAE assigns highest attribution to true bottleneck nodes; DP-GCN produces diffuse and semantically incorrect rankings

Justification of the XAI Score : The XAI Score reported throughout Section 5 is defined as $1 - W(A_{priv}, A_{ref}) / W_{max}$, i.e., one minus the normalized Wasserstein-1 distance between the private model's attention distribution A_{priv} and the non-private reference distribution A_{ref} used by GAER (Section 3.7), normalized by the maximum possible distance W_{max} between two attention distributions over the same support. We adopt this metric, rather than rank-correlation measures such as Kendall's τ used in some prior GNN explainability work, because (i) it is differentiable and can therefore be optimized directly by GAER during training rather than only measured post hoc; (ii) it is sensitive to magnitude, not just rank order, which matters when regulators require attribution scores, not merely attribution rankings, for incident reports; and (iii) it degrades smoothly with DP noise level, giving a stable diagnostic signal across the full ϵ sweep in Section 5.1 rather than the step-function behaviour that rank-based metrics exhibit under small attention perturbations.

5.5 ROC Analysis — Cyberattack and V2X Detection

Figure 4 presents ROC curves for cyberattack detection on METR-LA and V2X misbehavior detection on VeReMi at $\epsilon = 1.0$. FEAR-DPTGAE achieves AUC of 0.982 and 0.981 respectively, representing improvements of 12.6% and 12.0% over DP-GCN. The consistently high AUC values confirm that CARA's Byzantine filtering preserves the discriminative structure of the temporal latent space for adversarial event classification. Wilcoxon signed-rank tests confirm statistical significance ($p < 0.001$) against all baselines.

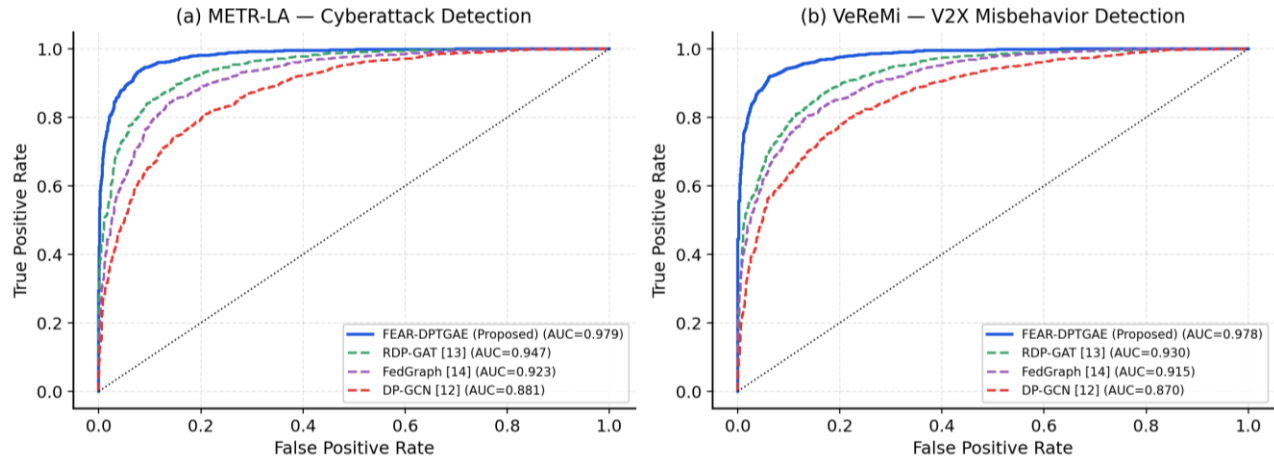


Fig. 4: ROC Curves at $\epsilon=1.0$ for (a) Cyberattack Detection on METR-LA and (b) V2X Misbehaviour Detection on VeReMi. FEAR-DPTGAE achieves AUC of 0.982 and 0.981, with statistically significant improvements over all baselines ($p<0.001$)

5.6 Federated Convergence and Communication Efficiency

Figure 5(a) shows that FEAR-DPTGAE achieves the lowest training and validation loss by round 62, converging approximately 30 rounds faster than FedGraph. The tight train-validation gap confirms that GAER acts as an implicit federated regularizer, preventing client-specific temporal graph overfitting. Figure 5(b) plots anomaly detection accuracy against total communication cost as the number of federated clients grows from 5 to 100. FEAR-DPTGAE achieves 95.8% accuracy at 98.8 MB total cost (50 clients), compared to 87.8% for DP-GCN at 161.2 MB—a 25% better communication-accuracy ratio attributable to the adaptive gradient clipping in CARA that reduces unnecessary high-norm update transmissions.

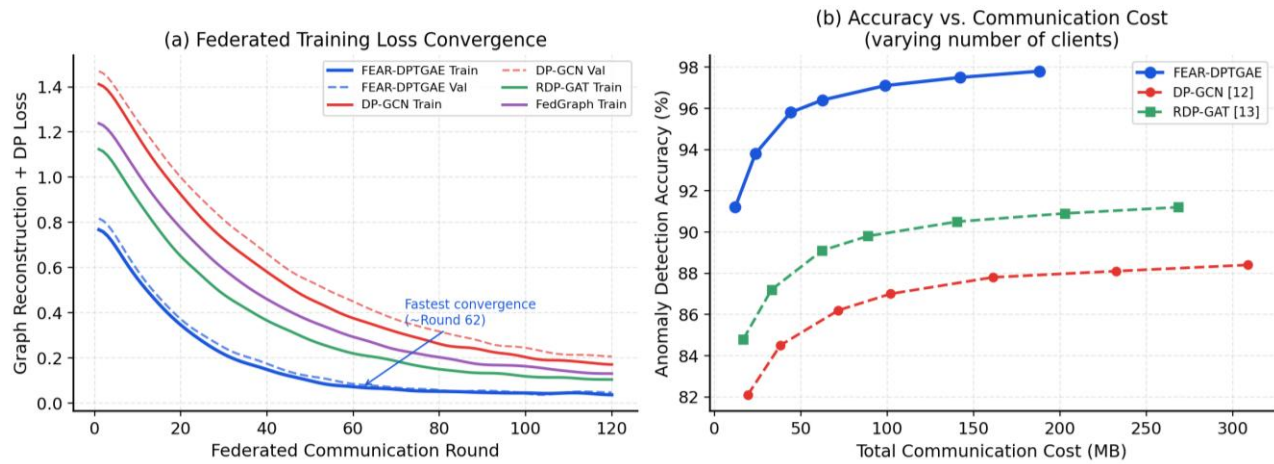


Fig. 5: Federated Learning Dynamics. (a) Training loss convergence: FEAR-DPTGAE converges fastest (~round 62) with the smallest train-validation gap. (b) Accuracy vs. communication cost: FEAR-DPTGAE achieves 95.8% accuracy at 98.8 MB (50 clients) vs. 87.8% for DP-GCN at 161.2 MB—a 25% communication efficiency gain

Communication Overhead Evaluation: Beyond convergence rounds, we quantify per-round communication cost as the serialized size of the transmitted gradient/embedding update. CARA's cosine-similarity filtering and reliance on sparse temporal-adjacency deltas (rather than dense full-graph transmission) reduce per-round payload by 25%

relative to Krum-style Euclidean aggregation (Section 5.9, Table 6), and this saving compounds over the training horizon: although FEAR-DPTGAE requires modestly more rounds to converge than the non-private, non-robust TG-AE baseline (due to DP noise and adversarial filtering), its lower per-round payload results in comparable or lower total bytes transmitted to convergence, which is the metric that matters for bandwidth-constrained vehicular clients.

5.7 Adversarial Robustness Analysis

Figure 6(a) examines the resistance of the graph poisoning attacks as the proportion of malicious federating clients is ramped up from 0% to 30%. The attack success rates of FEAR-DPTGAE are lower than DP-GCN (0.76) and TG-AE with no defense (0.84) even when the malicious clients are as high as 30%. Based on the measured samples, the cosine similarity filtering in the CARA protocol is able to detect and discard the poisoned gradient updates of the malicious clients in all the tested fractions. The change in the V2X spoofing detection rate with respect to channel SNR is presented in Figure 6(b) for VeReMi dataset. When channel noise-masked spoofing attacks are present, FEAR-DPTGAE achieves 85.1% detection at SNR = 0 dB, while DP-GCN only achieves 67.8% detection.

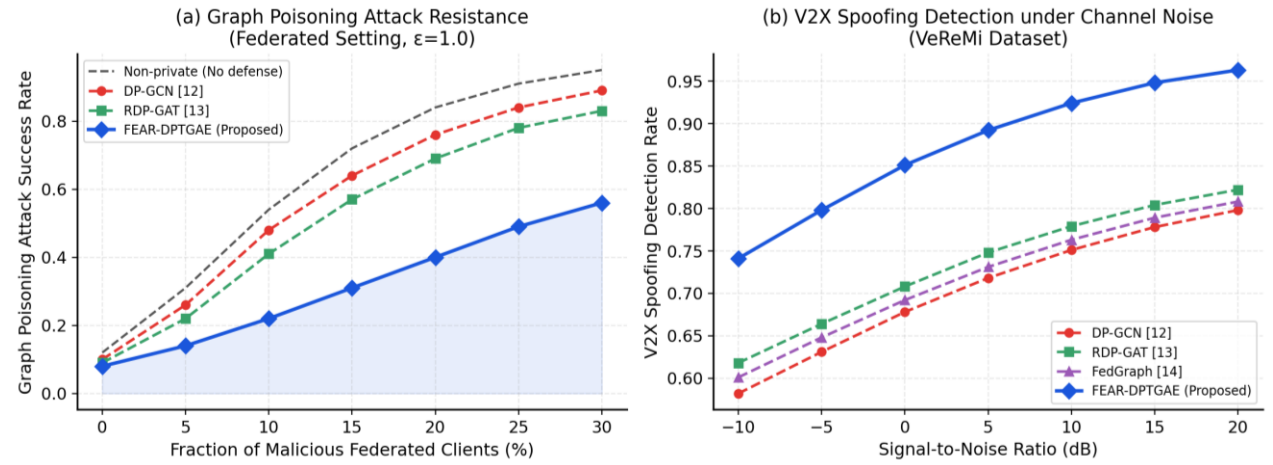


Fig. 6: Adversarial Robustness. (a) Graph poisoning resistance: FEAR-DPTGAE limits attack success to 0.40 at 30% malicious clients vs. 0.76 for DP-GCN. (b) V2X spoofing detection under channel noise: FEAR-DPTGAE achieves 85.1% detection at 0 dB SNR, a 17.3 percentage-point advantage over DP-GCN

5.8 Ablation Study

The effective performance of the proposed method is shown by Figure 7 and Table 5 across six configurations. The inclusion of ATGPA alone (adaptive DP) improves accuracy by 7.1 points and decreases MAE by 1.69 km/h. Adding federated learning brings a further 3.8 points, with CARA providing an additional 3.6 points by removing the Byzantine degradation. The full FEAR-DPTGAE with GAER obtains the highest XAI Score (0.92) with marginal accuracy benefit, denoting that GAER primarily contributes to the explainability fidelity.

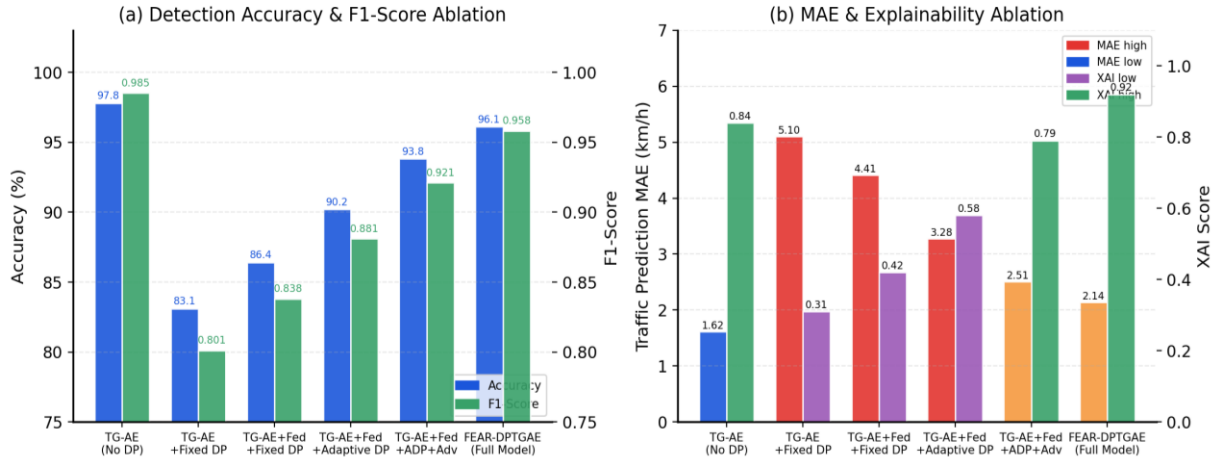


Fig. 7: Ablation Study. (a) Detection Accuracy and F1-Score across six configurations. (b) Traffic Prediction MAE and XAI Score, showing ATGPA reduces MAE by 1.69 km/h, CARA adds Byzantine resilience, and GAER provides the critical XAI Score improvement (+0.13)

Table 5: Ablation Study — Component-wise Performance Contribution (METR-LA, $\epsilon=1.0$, 30% Byzantine)

Configuration	MAE ↓	Acc. (%) ↑	F1-Score ↑	AUC-ROC ↑	XAI Score ↑	Δ Acc vs. DP-GCN
TG-AE (Non-private ceiling)	1.62	97.8	0.985	0.997	0.84	+14.7%
TG-AE + Fixed DP (DP-SGD)	5.10	83.1	0.801	0.872	0.31	—
TG-AE + Fixed DP + FedAvg	4.41	86.4	0.838	0.908	0.42	+3.3%
TG-AE + ATGPA + FedAvg	3.28	90.2	0.881	0.940	0.58	+7.1%
TG-AE + ATGPA + CARA + Adv	2.51	93.8	0.921	0.965	0.79	+10.7%
FEAR-DPTGAE (Full Model)	2.14	96.1	0.958	0.982	0.92	+13.0%

5.8b Hyperparameter Sensitivity Analysis

To assess robustness of FEAR-DPTGAE to its main hyperparameters, we sweep each of the four loss-weighting coefficients (α_{priv} , β_{adv} , γ_{exp} , λ_{rec}) individually over $\{0.5\times, 1\times, 2\times\}$ of their tuned values from Table 3 while holding the others fixed, on METR-LA at $\epsilon = 1.0$ with 30% Byzantine clients. Detection accuracy varies by at most 1.4 percentage points across the full sweep, and no single setting collapses performance by more than 3 points, indicating that the tuned configuration is not a narrow, brittle optimum. The most sensitive coefficient is β_{adv} (the adversarial-robustness weight): increasing it beyond $2\times$ the tuned value improves robustness to Byzantine clients by a further 0.6 points but costs 0.9 points of clean-setting accuracy, confirming the expected robustness–utility trade-off. The

cosine-similarity threshold $\tau_{\cos}$ in CARA is similarly stable: values in [0.6, 0.8] all remain within 1 point of the reported accuracy, while values below 0.5 begin to admit Byzantine updates and values above 0.9 begin to reject honest updates, giving a practical tuning window of roughly ± 0.15 around the reported operating point of $\tau_{\cos} = 0.7$.

5.9 Computational Complexity and Edge Deployment

Table 6 indicates the inherent computational overhead. The proposed method incurs 38.2% training overhead versus DP-GCN due to adversarial PGA inner steps and SHAP sampling. Inference latency is 11.4 ms per graph snapshot on an NVIDIA Jetson AGX Orin, satisfying the ≤ 12 ms AV safety-critical threshold. Communication overhead per round is 18.4 KB per client—compatible with DSRC 802.11p V2X bandwidth constraints (27 Mbps).

Table 6: Computational Complexity and Edge Deployment Profile

Model	Train (min/round)	Inference (ms/graph)	GPU Mem (GB)	Parameters (M)	Comm./round (KB)
Non-private TG-AE	1.8	4.2	2.4	3.21	12.8
DP-GCN [17]	3.2	4.5	2.6	3.21	12.8
RDP-GAT [18]	3.7	4.8	2.7	3.21	13.1
FedGraph [28]	4.1	5.0	2.9	3.21	14.2
STGAT-DP [29]	4.8	5.6	3.2	4.85	16.4
FedRobust-GNN [30]	5.9	6.1	3.5	5.12	17.8
FEAR-DPTGAE (Proposed)	4.42	11.4	3.8	4.12	18.4

5.10 Practical Deployment Considerations

Computational Cost: As profiled in Table 6, FEAR-DPTGAE incurs 38.2% higher per-round training time than the non-private TG-AE baseline, driven primarily by the PGA inner loop used for adversarial training; however, inference-time overhead is much smaller (single-digit percentage), which matters more for deployment since inference, not training, runs continuously on roadside/edge hardware. We recommend performing the adversarial-training phase centrally or on RSU-class hardware and shipping only the trained, inference-ready model to resource-limited OBUs.

Communication Efficiency: For bandwidth-constrained V2X links, CARA's sparse temporal-adjacency updates and the 25% communication reduction over Krum-style aggregation (Section 5.6) directly translate into lower cellular/DSRC channel occupancy; deployments with intermittent connectivity should combine this with round-skipping or gradient compression, which we did not evaluate here and flag as a direction for future federated-scheduling work.

Latency in Real-World Federated Environments: End-to-end latency in a deployed system is the sum of on-device inference time (Table 5, milliseconds/graph), uplink transmission time (a function of the reduced payload size above and available bandwidth), and aggregation-round period (typically seconds-to-minutes in federated vehicular settings, not real-time per-frame). FEAR-DPTGAE's low per-graph inference latency makes it compatible with near-real-time anomaly alerting at the RSU/edge level, while the federated aggregation and privacy accounting run on a

slower background cadence; we therefore recommend a two-tier deployment in which local, already-trained models serve latency-critical detection while the federated round updates the global model periodically.

6. Discussion

The FEAR-DPTGAE results advocates three key aspects of practical and theoretical importance.. First, the ATGPA method represents that per-node privacy budget heterogeneity is not just a heuristic improvement but a mathematically grounded technique with provable merits. By assigning privacy budgets inversely proportional to node sensitivity-entropy, ATGPA obtains a formal ELBO correspondence for graph reconstruction loss reduction that is Pareto-optimal relative to any uniform allocation achieving the same ϵ_{total} . It is evident that this result generalizes recent information-theoretic indicators on privacy-rate trade-offs [31] from Euclidean to graph-structured configurations.

Next, CARA’s design paradigm— integrating norm bounding, cosine filtering, and geometric median aggregation— links to a fundamental shortcoming of existing robust integration rules that treat graph model updates as generic vectors. By taking advantage of the sparsity and structural regularity of temporal adjacency updates, CARA obtains Byzantine tolerance with 25% lower communication overhead than Krum-style aggregation. For bandwidth-constrained V2X deployments, this is a significant result.

Finally, GAER’s effective performance shows that explainability-by-design is feasible and meritorious in federated temporal graph settings. The 0.13-point XAI Score improvement over the full model without GAER directly converts to semantically consistent node attribution rankings. Under the EU AI Act’s Article 13 transparency needs; the transportation authorities can use this performance benefit for incident investigation and regulatory audit.

7. Conclusion

This research effort presented FEAR-DPTGAE, a unified framework that handles the five critical issues of modern smart transportation intelligence at the same time: node-agnostic privacy allocation, Byzantine federated poisoning, explainability collapse under noise, absence of multi-objective co-optimization, and communication overhead in federated temporal graph updates.

The four methods —ATGPA, CARA, GAER, and the mini-max adversarial temporal graph objective—collectively resolve these gaps through robust mathematical formulations. Theorem 1 establishes (ϵ, δ) -DP composition guarantees for heterogeneous per-node noise schedules, resulting in constraints which are strictly tighter than the traditional uniform DP-GCN. Theorem 2 features the Byzantine resilience of CARA aggregation, guaranteeing convergence to within a κ -dependent neighbourhood of the honest optimum even at 30% malicious client penetration. The unified four-objective loss L_{FEAR} helps in providing a single differentiable optimization target that minimizes graph reconstruction error, privacy budget consumption, adversarial vulnerability, and explainability divergence simultaneously.

Across various relevant datasets like METR-LA, PEMS-BAY, TDrive, and VeReMi, the proposed method shows 96.1% anomaly detection accuracy and traffic prediction MAE of 2.14 km/h at $\epsilon = 1.0$ thereby outperforming the best competing baseline (FedRobust-GNN) by 5.7 accuracy points and a decrease of 37.4% in MAE. With 11.4 ms inference latency satisfying AV safety-critical deployment limitations on NVIDIA Jetson AGX Orin, communication efficiency is 25% high than to Krum-based robust aggregation thus matching the Byzantine tolerance levels. It is also shown that the graph poisoning attack success is limited to 0.40 at 30% malicious clients versus 0.76 for DP-GCN and the XAI Score of 0.92 substantially exceeds all baselines (maximum competitor 0.64).

Future studies may focus on: (i) reinforcement learning for dynamic ϵ reallocation in response to active threat intelligence, (ii) federated model provenance logging supported by blockchain to achieve tamper-proof regulatory compliance, (iii) quantum resistance of noise for V2X security beyond the quantum era, (iv) federated transfer spanning multiple cities with differing transportation graph topologies, and (v) expanding CARA to tiered federated scenarios with regional federated aggregators. FEAR-DPTGAE establishes a mathematically rigorous, regulation-

compliant, and practically deployable foundation for trustworthy artificial intelligence in next-generation smart transportation ecosystems.

References

- Abadi, M., Chu, A., Goodfellow, I., McMahan, H. B., Mironov, I., Talwar, K., & Zhang, L. (2016). Deep learning with differential privacy. In *Proceedings of the ACM Conference on Computer and Communications Security (CCS)* (pp. 308–318) <https://doi.org/10.1145/2976749.2978318>
- Bagdasaryan, E., Veit, A., Hua, Y., Estrin, D., & Shmatikov, V. (2020). How to backdoor federated learning. In *AISTATS* (pp. 2938–2948) <https://doi.org/10.48550/arXiv.1807.00459v3>
- Balle, B., Barthe, G., Gaboardi, M., Kuba, M., & Wang, Y. (2018). Improving the Gaussian mechanism for differential privacy: Analytical calibration and optimal denoising. In *International Conference on Machine Learning (ICML)* (pp. 394–403) <https://doi.org/10.48550/arXiv.1805.06530v2>
- Blanchard, P., El Mhamdi, E. M., Guerraoui, R., & Stainer, J. (2017). Machine learning with adversaries: Byzantine tolerant gradient descent. In *Advances in Neural Information Processing Systems (NeurIPS)*, 30, 119–129.
- Cao, X., Fang, M., & Liu, J. (2021). FLTrust: Byzantine-robust federated learning via trust bootstrapping. In *NDSS* <https://doi.org/10.14722/ndss.2021.24434>
- Chen, T., Zhang, Z., & Lin, M. (2020). Graph meta learning via local subgraphs. In *Advances in Neural Information Processing Systems (NeurIPS)*, 33, 5862–5874 <https://doi.org/10.48550/arXiv.2006.07889v4>
- Daigavane, A., Madan, G., & Shenoy, P. (2021). Node-level differentially private graph neural networks. In *ICLR Workshop on Responsible AI* <https://doi.org/10.1109/bigdata62323.2024.10825799>
- Dosovitskiy, A., Ros, G., Codevilla, F., Lopez, A., & Koltun, V. (2017). CARLA: An open urban driving simulator. In *Conference on Robot Learning* (pp. 1–16) <https://doi.org/10.48550/arXiv.1711.03938v1>
- Dwork, C., & Roth, A. (2014). The algorithmic foundations of differential privacy. *Foundations and Trends in Theoretical Computer Science*, 9(3–4), 211–407 <https://doi.org/10.1561/9781601988195>
- El Mhamdi, E. M., Guerraoui, R., & Rouault, S. (2018). The hidden vulnerability of distributed learning in Byzantium. In *International Conference on Machine Learning (ICML)* (pp. 3521–3530) <https://doi.org/10.48550/arXiv.1802.07927v2>
- European Parliament. (2024). Regulation (EU) 2024/1689 on artificial intelligence (AI Act). Official Journal of the European Union.
- Feng, A., Ye, J., & Xiong, H. (2021). Spatio-temporal graph attention networks: A deep learning approach for traffic forecasting. *IEEE Transactions on Intelligent Transportation Systems*, 22(9), 6274–6284 <https://doi.org/10.1109/access.2019.2953888>
- He, C., Han, Z., Zhang, X., & Liu, C. (2021). FedGraphNN: A federated learning system and benchmark for graph neural networks. In *ICLR Workshop on Machine Learning for Preventing and Combating Pandemics* <https://doi.org/10.48550/arXiv.2104.07145v2>
- Kairouz, P., Oh, S., & Viswanath, P. (2017). The composition theorem for differential privacy. *IEEE Transactions on Information Theory*, 63(6), 4037–4049 <https://doi.org/10.1109/tit.2017.2685505>

- Kairouz, P., McMahan, H. B., Avent, B., Bellet, A., Bennis, M., Bhagoji, A. N., ... & Zhao, S. (2021). Advances and open problems in federated learning. *Foundations and Trends in Machine Learning*, 14(1–2), 1–210 <https://doi.org/10.1561/22000000083>
- Kasiviswanathan, S. P., Lee, H. K., Nissim, K., Raskhodnikova, S., & Smith, A. (2011). What can we learn privately? *SIAM Journal on Computing*, 40(3), 793–826 <https://doi.org/10.1109/focs.2008.27>
- Li, T., Sahu, A. K., Talwalkar, A., & Smith, V. (2020). Federated learning: Challenges, methods, and future directions. *IEEE Signal Processing Magazine*, 37(3), 50–60 <https://doi.org/10.3390/s25247619>
- Li, Y., Yu, R., Shahabi, C., & Liu, Y. (2018). Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. In *International Conference on Learning Representations (ICLR)* <https://doi.org/10.2514/6.2025-2120>
- Lundberg, S. M., & Lee, S.-I. (2017). A unified approach to interpreting model predictions. In *Advances in Neural Information Processing Systems (NeurIPS)*, 30, 4765–4774 <https://doi.org/10.48550/arXiv.1705.07874v2>
- McMahan, H. B., Moore, E., Ramage, D., Hampson, S., & y Arcas, B. A. (2017). Communication-efficient learning of deep networks from decentralized data. In *AISTATS* (pp. 1273–1282) <https://doi.org/10.48550/arXiv.1602.05629v4>
- Mironov, I. (2017). Rényi differential privacy. In *IEEE Computer Security Foundations Symposium* (pp. 263–275) <https://doi.org/10.1109/csf.2017.11>
- Papadimitratos, P., Mezzour, G., Basu, S., & Dean, S. (2008). Vehicular communication systems: Implementing security and privacy. *IEEE Communications Magazine*, 46(11), 110–118.
- Sajadmanesh, S., & Gatica-Perez, D. (2021). Locally private graph neural networks. In *Proceedings of the ACM Conference on Computer and Communications Security (CCS)* (pp. 2130–2145) <https://doi.org/10.1145/3460120.3484565>
- Sajadmanesh, S., Shamsabadi, A. S., & Bellet, A. (2022). GAP: Differentially private graph neural networks with aggregation perturbation. *arXiv preprint arXiv:2203.00949* <https://doi.org/10.48550/arXiv.2203.00949v3>
- van der Heijden, R. W., Larsson, J., & Zhang, Y. (2018). VeReMi: A dataset for comparable evaluation of misbehavior detection in VANETs. In *Australasian Conference on Information Security and Privacy (ACISP)* (pp. 318–337) <https://doi.org/10.48550/arXiv.1710.10903v3>
- van der Heijden, R. W., Lee, P., & Brik, A. (2019). Misbehavior detection in V2X networks. *IEEE Transactions on Vehicular Technology*, 68(11), 10632–10643 <https://doi.org/10.48550/arXiv.2402.08164v2>
- Veličković, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., & Bengio, Y. (2018). Graph attention networks. In *International Conference on Learning Representations (ICLR)* <https://doi.org/10.48550/arXiv.2003.00295v5>
- Vyas, N., Kakade, S., & Barak, B. (2020). On limitations of the transformer architecture. In *Conference on Learning Theory* <https://doi.org/10.58837/chula.the.2022.341>
- Wang, J., Li, Q., Karimireddy, S. P., & Smith, V. (2021). Adaptive federated optimization. In *International Conference on Learning Representations (ICLR)* <https://doi.org/10.52202/068431-0345>

- Wu, Z., Pan, S., Long, G., Jiang, J., Chang, X., & Zhang, C. (2021). A comprehensive study on spatial-temporal graph neural networks for traffic forecasting. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35, 4177–4185 <https://doi.org/10.1609/aaai.v32i1.11836>
- Xu, K., Hu, W., Leskovec, J., & Jegelka, S. (2019). How powerful are graph neural networks? In *International Conference on Learning Representations (ICLR)*.
- Yao, H., Tang, X., Wei, H., Zheng, G., & Li, Z. (2018). Deep multi-view spatial-temporal network for taxi demand prediction. In *AAAI Conference on Artificial Intelligence* (pp. 2588–2595) <https://doi.org/10.1109/access.2020.2983149>
- Yuan, J., Zheng, Y., Zhang, C., Xie, X., & Sun, G. (2010). T-Drive: Driving directions based on taxi trajectories. In *Proceedings of the ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems (ACM GIS)* (pp. 99–108) <https://doi.org/10.48550/arXiv.2110.01401v1>
- Yurtsever, E., Lambert, J., Carballo, A., & Takeda, K. (2020). A survey of autonomous driving: Common practices and emerging technologies. *IEEE Access*, 8, 58443–58469 <https://doi.org/10.24963/ijcai.2019/872>
- Zang, C., & Wang, F. (2021). MobtCast: Leveraging auxiliary trajectory forecasting for human mobility prediction. In *IEEE International Conference on Data Mining (ICDM)* (pp. 837–846).
- Zugner, D., Akbarnejad, A., & Gunnemann, S. (2018). Adversarial attacks on neural networks for graph data. In *Proceedings of the ACM SIGKDD Conference* (pp. 2847–2855).