Data Science and Engineering (2026) 11:199–212
https://doi.org/10.1007/s41019-025-00321-x

RESEARCH PAPERS

Generating Counterfactual Temporal Motifs: Unraveling the Mysteries of Temporal Graph Neural Networks

Yibowen Zhao[1] · Yonghui Xu[1] · Ning Liu[1] · Lizhen Cui[1] · Qingzhong Li[1]

Received: 10 May 2025 / Revised: 1 September 2025 / Accepted: 22 September 2025 / Published online: 24 January 2026

Abstract
Temporal Graph Neural Networks (TGNNs) are increasingly applied in dynamic scenarios, however, their limited explainability hinders their adoption in high-stakes domains. Existing methods tend to conflate causality with temporal proximity, leading to ambiguous explanations that mix impactful and irrelevant events. Moreover, they lack counterfactual reasoning to assess whether altering specific temporal events would change TGNN predictions. To overcome these challenges, we propose CTM-Explainer, which identifies critical temporal dependencies through iterative "what-if" perturbation analysis. To the best of our knowledge, this is the first post-hoc counterfactual explanation framework for TGNN. It enables precise attribution of how specific timestamped events influence TGNN predictions. By embedding causal analysis into a reinforcement learning framework, CTM-Explainer constructs Counterfactual Temporal Motifs (CTMs) that are causally grounded in model outcome shifts via interventional probability estimation. This design eliminates temporally correlated but non-essential events, while preserving those with verified causal influence. Extensive experiments on real-world and synthetic datasets confirm that CTM-Explainer generates more faithful and concise explanations than existing methods, at significantly lower computational cost.

Keywords Explainable Graph Neural Network · Temporal Graph Neural Network · Post-hoc Explanation · Graph Neural Network

1 Introduction

Temporal Graph Neural Networks (TGNNs) are specifically designed to model the dynamics of evolving graphs by integrating temporal dependencies, enabling them to capture complex spatio-temporal patterns in applications ranging from social network evolution to urban computation [1, 2]. Despite their predictive power, the opacity of TGNNs' decision-making severely limits their adoption in high-stakes domains such as clinical diagnostics and pharmaceutical risk assessment [3–5], where understanding prediction rationales is ethically imperative [6, 7]. The inherent complexity of temporal interactions–characterized by nonlinear event dependencies and dynamically shifting graph topologies [8]–poses unique challenges to explainability [9, 10]. This gap underscores the urgent need for solutions that integrate causal reasoning with temporal dynamics to deliver truly actionable insights.

Existing explanation frameworks predominantly rely on correlation-driven techniques, such as the Graph Information Bottleneck [11–13] and the heuristic temporal motif extraction[14], to align explanations with input graph structures via temporally contiguous subgraphs. However, by privileging temporal adjacency over causal necessity, these methods indiscriminately highlight both truly causal interactions, such as Specific Inquiries that directly influence predictions, and spurious temporal neighbors, such as

✉ Yonghui Xu
xu.yonghui@hotmail.com

✉ Lizhen Cui
clz@sdu.edu.cn

Yibowen Zhao
ybw.zhao@mail.sdu.edu.cn

Ning Liu
liun21cs@sdu.edu.cn

Qingzhong Li
lqz@sdu.edu.cn

1 School of Software & Joint SDU-NTU Centre for Artificial Intelligence Research, Shandong University, Jinan 250100, China

Generic Shares or Off-topic Replies that merely co-occur in time, when explaining a TGNN's prediction that a user will join a high-risk organization (Fig. 1)[15]. As a result, explanations become cluttered with redundant and imprecise events. In contrast, causal explanation models focus exclusively on factors with demonstrated causal influence, producing concise, high-quality explanations that effectively filter out coincidental correlations [16–18].

Moreover, prevailing post-hoc explainers provide only factual summaries of observed model behavior and cannot evaluate how predictions would respond to hypothetical changes. For instance, stakeholders may wish to determine whether modifying a user's Specific Inquiry at timestamp t_2 (the red edge in Fig. 1) would reduce the risk of joining the organization. Without counterfactual analysis, such "what-if" queries remain unanswered. A causal explanation framework, however, can perform rigorous counterfactual interventions [19, 20], isolating the exact event whose modification would flip the TGNN's outcome. This capability yields actionable insights, pinpointing the single change required to alter a prediction, which is indispensable in dynamic environments where transient interactions otherwise inflate explanations with irrelevant events [21, 22].

To address the critical challenges in TGNNs' explainability, this paper introduces CTM-Explainer, a causal counterfactual framework that reveals why and how temporal events drive TGNNs' outcomes. To transcend the limitations of factual reasoning, we propose a reinforcement learning paradigm that dynamically constructs Counterfactual Temporal Motifs (CTMs) through iterative causal-temporal exploration. To eliminate the conflation of correlation and causation, CTM-Explainer embeds Individual Causal Effect (ICE) quantification into its reward design, ensuring only events with validated causal significance are retained. When evaluated on real-world and synthetic datasets, our framework provides explainable insights for TGNNs like Temporal Graph Network (TGN) [23] and Temporal Graph Attention Network (TGAT) [24]. The key innovations include:

- To the best of our knowledge, this is the first post-hoc counterfactual explanation model for TGNNs that explicitly identifies causally influential events affecting prediction outcomes.
- By incorporating ICE into the reinforcement learning framework, CTM-Explainer generates CTMs that are causally grounded in TGNNs' outcome shifts through quantitative interventional probability analysis, thereby eliminating temporally correlated but non-essential events while preserving verified causal dependencies.
- Comprehensive evaluations demonstrate that CTM-Explainer achieves superior explainability metrics across diverse domains, while maintaining competitive computational efficiency against state-of-the-art TGNN explanation models.

The paper is structured as follows: Section 2 reviews existing factual and counterfactual explanation methods in temporal graphs. Section 3 formalizes Counterfactual Temporal Motifs and details the CTM-Explainer architecture. Section 4 presents comparative experiments. Section 5 concludes with broader implications.

2 Related Works

This section examines explanatory approaches for GNNs, with a specific focus on TGNNs. It reviews the progression of both factual and counterfactual explanatory models, emphasizing their development and limitations.

2.1 Explanatory Models for Static GNNs

To enhance explainability in graph neural networks, a broad range of post-hoc explanation methods have been developed for static graphs. One seminal approach is GNNExplainer [25], which identifies predictive subgraphs and node features by learning edge-level masks via perturbation. Despite its pioneering role, this perturbation process may introduce artifacts misaligned with the model's true reasoning. To address this, PGExplainer [26] introduces a learnable probabilistic framework that generates edge masks through mutual information maximization, improving explanation stability and alignment. Beyond perturbation-based techniques, some methods adopt surrogate modeling.

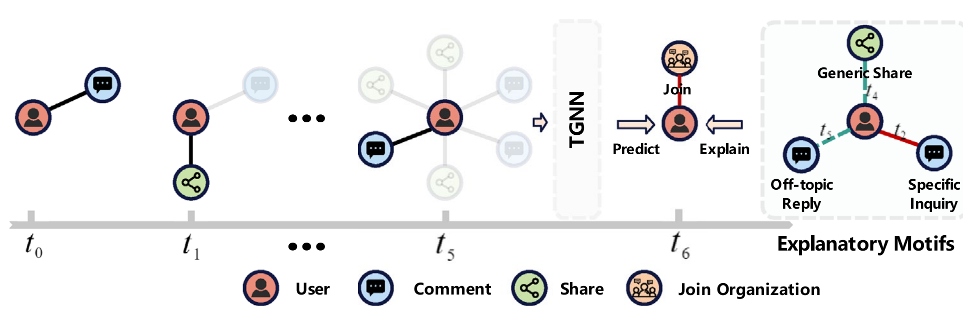


Fig. 1 Illustration of factual versus causal reasoning: while all three interactions (Generic Share, Specific Inquiry, Off-topic Reply) may appear in factual explanations, only the Specific Inquiry is causally responsible for the TGNN prediction

PGM-Explainer [27], for example, builds Bayesian networks to capture dependencies between input features and predictions, enabling conditional-probability-based explanations. This enhances explainability but often lacks flexibility across diverse real-world datasets.

In specialized domains like chemistry, SME [28] tailors explanation strategies by segmenting molecular graphs into chemically meaningful components, evaluating their individual contributions. While effective for molecule property prediction, its reliance on predefined substructures reduces transferability to other fields. To improve causal explainability, FANS [29] evaluates both the necessity and sufficiency of features by crafting counterfactual variants and estimating their effect on model outputs. However, this approach may underperform in heterogeneous or structurally diverse graphs due to its simplified perturbation schema. To capture fine-grained graph-level dependencies, SAME [30] explores subgraph structures via Monte Carlo tree search and approximates Shapley values to attribute prediction importance. Yet, its combinatorial complexity raises scalability concerns on large graphs. GNNShap [31] extends Shapley-based explanation with parallelized sampling and edge-level scoring, offering better scalability.

While these methods focus on explaining existing predictions, they often fall short in addressing what-if scenarios, thus motivating the development of counterfactual approaches for more actionable insights. This gap motivates the development of counterfactual approaches. Counterfactual models aim to enhance trust and transparency by identifying minimal input changes that alter GNN predictions. CF-GNNExplainer [32] pioneers this field, focusing on minimal graph alterations through binary masking and optimized loss functions. Chhablani et al. [33] advance this work by introducing game-theoretic semivalues, such as Shapley and Banzhaf values, offering theoretically grounded explanations beyond correlation-based methods. Robustness in counterfactual explanations is critical due to noise in real-world data. Robust Counterfactual Witnesses (RCWs) [34] introduces algorithms to maintain stability amidst graph modifications. However, RCWs primarily emphasize statistical correlations, often neglecting causal relationships. Bajaj et al. [35] address this gap by leveraging common decision logic to establish causal links between explanations and GNN outcomes, improving noise resilience. For unsupervised GNNs, UNR-Explainer [36] identifies critical subgraphs affecting k-nearest neighbors, enabling insights for unlabeled data tasks. Despite its utility, its heavy parameterization reduces efficiency. InduCE [37] mitigates this by employing reinforcement learning to strategically select perturbations, accelerating counterfactual graph generation. InduCE also broadens the perturbation scope by incorporating edge additions, enriching the explanation process.

Nevertheless, these static GNN explanatory models predominantly focus on structural dependencies and lack the capability to handle temporally evolving graphs, limiting their applicability to TGNNs.

2.2 Factual-reasoning Explanatory Models for TGNNs

In TGNNs, Xia et al. [15] propose T-GNNExplainer, which uses a Monte Carlo Tree Search-powered Explorer to identify event subsets and a pretrained Navigator to guide the search. Despite its innovation, T-GNNExplainer struggles with the inherent complexity of temporal graph explanations. To address these challenges, TempME [14] uncovers temporal motifs and uses null models to evaluate information content, emphasizing sparsity and simplicity. However, it faces constraints when applied to heterogeneous temporal graphs. Li et al. [38] extend this line of research with HTGExplainer, which reparameterizes subgraph generation using a neural network and incorporates heterogeneous and temporal edge embeddings, achieving rational explanations for heterogeneous temporal GNNs.

While these models capture temporal dependencies to some extent, they focus on factor-based reasoning, often yielding imprecise explanations that fail to clarify TGNN decision-making processes. Such methods lack the ability to pinpoint minimal feature changes that drive prediction shifts, which are essential for actionable insights.

3 Methodology

This section provides an in-depth exploration of the CTM-Explainer, a novel TGNN's counterfactual explanatory model. The CTM-Explainer is designed to address the complexities of temporal graph analysis by introducing a new concept: Counterfactual Temporal Motif. To ensure a comprehensive understanding, this section starts with an overview of temporal graphs and their dynamic nature, followed by a formal definition of CTM. This section then delves into the causal analysis powered by Individual Causal Effect (ICE), which is essential for generating concise and impactful CTM that directly influence the TGNNs' outcomes. Finally, the section concludes with a presentation of the CTM-Explainer's systematic approach to generating CTM, as depicted in Fig. 2.

3.1 Definitions

Unlike static graphs represented as $G = \{V, E\}$, temporal graphs incorporate temporal dimensions to capture network

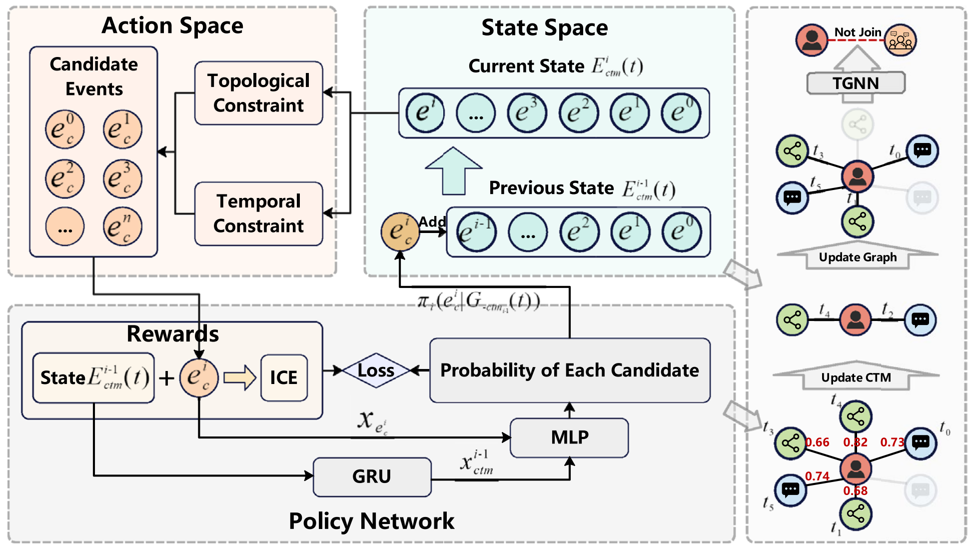


Fig. 2 The overall framework of CTM-Explainer, which formulates counterfactual explanation as a Markov Decision Process. At each step, the explainer selects candidate events based on spatio-temporal constraints and receives ICE-based rewards to iteratively construct a compact and causally-grounded Counterfactual Temporal Motif

dynamics effectively. Based on prior work [15], we formally define a temporal graph as follows:

Definition 1 (Temporal Graph) A temporal graph models a network's evolution via a sequence of timestamped interactions. Formally:

$$\mathcal{G}(t) = \{V(t), E(t)\}, \tag{1}$$

where *V*(*t*) denotes the node set at or before timestamp *t*, and *E*(*t*) denotes the set of events occurring before *t*. Each event $e_k \in E(t)$ is defined as:

$$e_k = (u_k, v_k, t_k, a_k), \tag{2}$$

indicating an interaction between nodes u_k and v_k at timestamp t_k with attribute a_k. Interactions are considered undirected.

To facilitate precise and actionable explanations in temporal contexts, we introduce the Counterfactual Temporal Motif (CTM). Ideally, counterfactual explanations aim to achieve minimal changes to alter predictions; however, due to practical constraints in real-world scenarios, we typically fix the size of explanations and optimize for maximum impact.

Definition 2 (Counterfactual Temporal Motif, CTM) Given a trained TGNN $f(\cdot)$ that predicts label $\hat{y}$ for target event e^t with probability $P(\hat{y}|\mathcal{G}(t))$, a CTM is a set $E_{\text{ctm}}(t) = e_1, \ldots, e_n$ where each element corresponds to either: (i) Removal of an existing event from $\mathcal{G}(t)$, or (ii) Addition of a synthetic event $\tilde{e}_i = (u_i, v_i, t_i, a_i)$ with $|t_i - t_{e^t}| \leq \tau$ and nodes u_i, v_i present at t_i. The counterfactual graph $\mathcal{G}_{\text{ctm}}(t)$ is obtained by applying these modifications. The CTM maximally reduces confidence in the original prediction:

$$E^*_{\text{ctm}}(t) = \underset{E_{\text{ctm}}(t)}{\arg\max}\ \Delta P, \tag{3}$$

$$\text{where } \Delta P = P(\hat{y}|\mathcal{G}(t)) - P(\hat{y}|\mathcal{G}_{\text{ctm}}(t)), \tag{4}$$

and ΔP is evaluated only when

$$\arg\max_{y} P(y \mid \mathcal{G}_{\text{ctm}}(t)) \neq \hat{y}, \tag{5}$$

ensuring that the counterfactual modification shifts the prediction towards an alternative label. This optimization is performed under sparsity constraints that fix the size of $E_{\text{ctm}}(t)$ to n. $E_{\text{ctm}}(t)$ denotes the set of CTM modifications; $\mathcal{G}_{\text{ctm}}(t)$ denotes the graph after applying the CTM modifications; $P(\hat{y} \mid \mathcal{G})$ denotes the probability of the original prediction; ΔP denotes the reduction in the original prediction probability; and τ denotes the temporal constraint.

CTM-Explainer leverages causal analysis to ensure concise, theoretically sound generation of CTMs under these practical considerations.

3.2 Causal Analysis for CTM Generation

The core innovation of CTM-Explainer lies in its systematic integration of causal reasoning, particularly the Individual Causal Effect (ICE), into the generation of counterfactual explanations, enabling concise and causally meaningful interpretations for Temporal Graph Neural Networks (TGNNs). Compared with existing approaches, such as Granger causality [39], which infers causal relationships mainly from lagged correlations and thus often produces overly broad explanations containing redundant or irrelevant events, and backdoor-adjustment methods [40], which rely heavily on explicitly identifying and controlling confounders and are thus impractical in dynamically evolving temporal networks, ICE directly quantifies the causal impact of individual events. This allows CTM-Explainer to identify events that are not only correlated with, but also causally responsible for, the TGNNs' outcomes.

To systematically quantify the causal influence of events in temporal graphs, we follow [41] and introduce causal interventions via the do-operator. The primary motivation for using such interventions is to explicitly cut off indirect causal paths and potential confounding interactions, thereby isolating the direct effect of each event. Importantly, before performing interventions, we pre-construct a candidate event set for each target event based on strict temporal and topological constraints (Eq. 13). This step localizes the explanatory scope to a bounded spatio-temporal subgraph containing only events within k hops and τ timestamps of the target. As a result, the original complex temporal dependency problem is weakened into a smaller, locally ordered and causally sufficient structure, in which long-range feedback loops and cross-window dependencies are substantially reduced. This transformation allows the subgraph to be approximately treated as a static DAG for the purpose of applying the do-calculus, avoiding the violations that would arise if interventions were attempted on the full-scale temporal network.

Formally, given a temporal graph $\mathcal{G}(t)$ representing all events up to timestamp t, we denote by $\mathcal{G}_{\text{ctm}}(t)$ the graph after modifying with the selected CTM. To further examine the causal significance of a single event e_c, we define $\mathcal{G}_{\text{ctm}}^{e_c}(t)$ as the graph obtained by additionally modifying e_c from $\mathcal{G}_{\text{ctm}}(t)$. The ICE for a single instance is:

$$\text{ICE}(e_c \mid \mathcal{G}_{\text{ctm}}(t)) = P(\hat{y} \mid \mathcal{G}_{\text{ctm}}(t)) - P(\hat{y} \mid \mathcal{G}_{\text{ctm}}^{e_c}(t)). \tag{6}$$

Equation 6 quantifies the causal effect of e_c by comparing TGNN predictions before and after the intervention. When averaging over the distribution of temporal graphs $p(\mathcal{G})$, this becomes:

$$\mathbb{E}_{p(\mathcal{G})}\left[P(\hat{y} \mid \mathcal{G}_{\text{ctm}}(t)) - P(\hat{y} \mid \mathcal{G}_{\text{ctm}}^{e_c}(t))\right]. \tag{7}$$

Although Eq. 6 provides a clear causal quantity at the instance level, direct computation of ICE in temporal graphs can be unstable and costly, since it requires repeated forward passes for each candidate event and only reflects probability changes for the single predicted label $\hat{y}$. To obtain a more robust and theoretically grounded measure, we move from the instance-level view of ICE to a distributional perspective. ICE evaluates the probability change of the predicted label $\hat{y}$ for a single instance, whereas mutual information extends this notion to the entire predictive distribution $p(y \mid \mathcal{G})$, thereby generalizing the instance-level effect of ICE to a distribution-level dependency between predictions and input subgraphs.

The key observation is that ICE essentially quantifies how much the predictive distribution is altered when e_c is intervened upon. Instead of restricting attention to the single probability $P(\hat{y} \mid \mathcal{G})$, we can characterize the overall dependency between predictions and the input subgraph using mutual information. Formally, the statistical dependency between the TGNN output random variable Y and the input subgraph $\mathcal{G}_{\text{ctm}}(t)$ is defined as

$$I(Y; \mathcal{G}_{\text{ctm}}(t)) = H(Y) - H(Y \mid \mathcal{G}_{\text{ctm}}(t)), \tag{8}$$

where $H(Y) = -\sum_{y\in\mathcal{Y}} p(y) \log p(y)$ denotes the entropy of predictions and $H(Y \mid \mathcal{G}) = -\sum_{y\in\mathcal{Y}} p(y \mid \mathcal{G}) \log p(y \mid \mathcal{G})$ the conditional entropy after observing $\mathcal{G}$.

If e_c is modified, the corresponding subgraph becomes $\mathcal{G}_{\text{ctm}}^{e_c}(t)$, and the drop in mutual information is

$$\Delta I(e_c) = I(Y; \mathcal{G}_{\text{ctm}}(t)) - I(Y; \mathcal{G}_{\text{ctm}}^{e_c}(t)). \tag{9}$$

This quantity captures the reduction in predictive information caused by removing (or adding) e_c. Intuitively, if e_c is causally important, its intervention decreases the amount of information the subgraph provides about Y.

We therefore adopt $\Delta I(e_c)$ as a distributional surrogate for ICE and denote it by $\hat{\text{ICE}}(e_c)$. Unlike Eq. 6, which focuses only on the probability of the predicted label $\hat{y}$, this surrogate evaluates the divergence between the full predictive distributions before and after the intervention, providing greater numerical stability while preserving the relative ranking of candidate events. By expressing the distributional surrogate in entropy form, we obtain the following equivalent formulation:

$$\hat{\text{ICE}}(e_c) = H(Y \mid \mathcal{G}_{\text{ctm}}^{e_c}(t)) - H(Y \mid \mathcal{G}_{\text{ctm}}(t)), \tag{10}$$

showing that ICE directly corresponds to the increase in predictive uncertainty after removing or adding e_c. Expanding this using TGNN predictive probabilities gives:

$$\hat{\text{ICE}}(e_c) = \sum_{y\in\mathcal{Y}} p(y \mid \mathcal{G}_{\text{ctm}}(t)) \log \frac{p(y \mid \mathcal{G}_{\text{ctm}}(t))}{p(y \mid \mathcal{G}_{\text{ctm}}^{e_c}(t))}. \tag{11}$$

i.e., ICE can be seen as a divergence between predictive distributions before and after modifying e_c. Finally, we select the CTM by solving:

$$\max_{E_{\text{ctm}}(t)} \sum_{e_c\in E_{\text{ctm}}(t)} \hat{\text{ICE}}(e_c \mid \mathcal{G}_{\text{ctm}_{i-1}}(t)), \quad \text{s.t.} \quad |E_{\text{ctm}}(t)| = n, \tag{12}$$

where $\mathcal{G}_{\mathrm{ctm}_{i-1}}(t)$ denotes the graph state after $i-1$ modifications. In practice, we solve Eq. 12 with a reinforcement learning agent that iteratively selects the event maximizing ICE at each step, yielding a compact and causally meaningful CTM.

3.3 CTM-Explainer

The following section introduces the key component of the proposed CTM-Explainer, which builds upon the theoretical analysis presented in the previous sections. Designed specifically as a counterfactual, post-hoc explanatory model for TGNNs, CTM-Explainer aims to identify critical input events whose modification substantially alters the outcomes of TGNN. To address computational complexity concerns, CTM-Explainer adopts a streamlined and efficient generation process formulated within a Markov Decision Process framework. Specifically, the explanatory procedure is systematically decomposed into four fundamental components: the state space *S*, the action space *A*, the reward function *R*, and the policy network π. Together, these components collaboratively guide the incremental construction of CTMs, ensuring that the explanations maintain temporal consistency and causal relevance. The remainder of this section provides detailed operational descriptions of these core components and illustrates how CTM-Explainer enhances the explainability of TGNN.

State Space (*S*) The Markov Decision Process framework is initialized with a state space (*S*) that begins as an empty set, s_0, and evolves to encapsulate the growing CTM, $E_{\mathrm{ctm}_i}(t)$, representing the aggregation of selected explanatory events up to the i-th step. The sequence of states, $S = \{s_0, s_1, ..., s_n\}$, mirrors the construction of the CTM, reflecting the dynamic evolution of the temporal graph as events are incrementally integrated.

Action Space (*A*) The action space (*A*) at each step i, A_i, comprises the set of candidate counterfactual modification that could be applied to the temporal graph. Candidate modifications are selected based on topological and temporal proximity to the target event e_t, adhering to spatio-temporal plausibility constraints:

$$\begin{aligned} &\mathrm{dist}(e_t, e_i) \le k, \\ &\quad |t_{e_t} - t_i| \le \tau, \\ &\qquad u_i, v_i \in V(t_i) \quad \text{(for addition operations)} \end{aligned} \tag{13}$$

where $\mathrm{dist}(\cdot)$ denotes the shortest path distance in $\mathcal{G}(t)$, $k = 2$ is the hop limit, $\tau = 2$ is the temporal threshold, and $V(t_i)$ ensures nodes exist at the intervention time. Synthetic events satisfying these constraints are considered valid candidates.

Rewards (*R*) The rewards metric is pivotal in quantifying the causal efficacy of each candidate event e_c^i within the context of the emergent CTM. The reward R_i is a direct reflection of the ICE of the modification of event based on the $\mathcal{G}_{\mathrm{ctm}_{i-1}}(t)$, serving as a catalyst for the selection of events with the most substantial explanatory power, formalized as:

$$R(e_c^i|\mathcal{G}_{\mathrm{ctm}_{i-1}}(t)) = \mathrm{I\hat{C}E}(e_c^i|\mathcal{G}_{\mathrm{ctm}_{i-1}}(t)). \tag{14}$$

Policy Network (π) The policy network, π, is the linchpin of Markov Decision Process framework, dictating the selection probabilities of actions given the current state. It harnesses the power of deep networks, e.g Gated Recurrent Unit (GRU) or Multilayer Perceptron (MLP), to encode the sequence representation x_{ctm}^{i-1} of the $E_{\mathrm{ctm}_{i-1}}(t)$,

$$\hat{x}_{\mathrm{ctm}}^{i-1} = \mathrm{GRU}(x_{\mathrm{ctm}}^{i-1}), \tag{15}$$

followed by a MLP to estimate the probability of selecting a candidate event e_c^i, expressed as:

$$\pi_i(e_c^i|\mathcal{G}_{\mathrm{ctm}_{i-1}}(t)) = \sigma(\mathrm{MLP}([x_{e_c^i} \| \hat{x}_{\mathrm{ctm}}^{i-1}])), \tag{16}$$

where σ denotes the sigmoid function, ensuring the valid probability. In this paper, the sequence representation x_{ctm}^{i-1} is obtained by concatenating the representations of all events in $E_{\mathrm{ctm}_{i-1}}(t)$. The representation for an individual event e is constructed using the following concatenation formula:

$$x_e = \mathrm{CAT}(x_u; x_v; x_t; a), \tag{17}$$

where x_u and x_v denote the source and destination node representations, x_t represents the temporal embedding encoded using a harmonic encoder, and a denotes the original attributes of the event. Furthermore, we use the representation of the target event as x_e at initial stage.

The policy network is refined through a policy gradient-based training regimen that aims to maximize the expected reward, thereby encouraging the selection of actions with high causal contributions to the explanatory CTM and the TGNN's outcomes. The optimization is formalized by:

$$\max_\theta \mathbb{E}_{\mathcal{G}_{\mathrm{ctm}}(t)\sim O}\left[R(e_c^i|\mathcal{G}_{\mathrm{ctm}_{i-1}}(t)) \log P_\theta(e_i^c|\mathcal{G}_{\mathrm{ctm}_{i-1}}(t))\right], \tag{18}$$

where θ denotes the policy network parameters, O denotes the graph distribution, $R(e_c^i|\mathcal{G}_{\mathrm{ctm}_{i-1}}(t))$ denotes the reward for the action in the given graph state, and $P_\theta(e_i^c|\mathcal{G}_{\mathrm{ctm}_{i-1}}(t))$ denotes the probability of selecting the action predicated on the current state. The training of the policy network is presented in Algorithm 1.

Input: Temporal Graph $\mathcal{G}(t)$, TGNN $f(\cdot)$, Target Event e_t
Output: CTM $E_{\text{ctm}}(t)$
for *each training epoch* **do**
  Initialize CTM $E_{\text{ctm}_0}(t)$ for e_t
  for *each generation step* **do**
    Construct action space using Eq. (13)
    Compute reward using Eq. (14)
    Obtain x_{ctm}^{i-1} via Eq. (15)
    Compute policy $\pi_i(e_c^i|\mathcal{G}_{\text{ctm}_{i-1}}(t))$
    Use Eq. (16) to get selection probability
    Execute action to get $E_{\text{ctm}_i}(t)$
    Compute loss by Eq. (18)
  end
  Compute average loss and perform backpropagation
  Record $E_{\text{ctm}}(t)$
end

Algorithm 1 Training of CTM-Explainer

3.4 Computational Complexity and Scalability Analysis

We propose **CTM-Explainer**, a reinforcement learning-based framework for TGNNs, designed to significantly enhance computational efficiency and scalability. Compared to T-GNNExplainer, which relies on Monte Carlo Tree Search with complexity $O\left(R \times \left(C \times \log C + C + LF \times \bar{d}\right)\right)$, where LF denotes the number of TGNN forward passes, and $\bar{d}$ is the average node degree. The candidate set size is defined as $C = N \times \bar{d} \times \Delta T$, where N is the number of nodes and ΔT is the temporal window size. CTM-Explainer substantially reduces computational overhead. Specifically, T-GNNExplainer's complexity scales significantly with the candidate set size C and the number of roll-out iterations R, making it computationally intensive for large graphs. CTM-Explainer achieves lower complexity $O\left(K \times \left(C + HF + LF \times \bar{d}\right)\right)$, where HF represents the number of hidden-layer operations in the CTM-Explainer's policy network, and K is the number of reinforcement learning iterations. By limiting dependency on candidate size C and focusing on stable learning iterations K, CTM-Explainer enables efficient action prediction via neural networks (e.g., GRU or MLP).

Overall, CTM-Explainer's streamlined and scalable design effectively addresses the computational bottlenecks in interpreting predictions on large-scale temporal graph datasets, making it suitable for practical deployment in real-world systems that require both predictive accuracy and explainability. For example, in an urban traffic forecasting scenario, where TGNNs are used to predict congestion based on vehicle movements over time, the temporal graph may include thousands of timestamped events within short intervals. Traditional methods such as T-GNNExplainer struggle in this context due to the exponential cost of Monte Carlo Tree Search over large candidate sets. In contrast, CTM-Explainer incrementally constructs a minimal set of counterfactual events–such as specific vehicle interactions within critical time windows–that, if removed, would change the TGNN's prediction from "congested" to "not congested". This process is guided by a reinforcement learning policy (see Section 3.3) and quantified through ICE-based causal analysis (see Section 3.2), ensuring that each selected event is both computationally efficient and causally meaningful. As a result, CTM-Explainer enables effective and interpretable decision support in time-sensitive domains such as traffic management, financial risk assessment, and clinical event monitoring.

4 Experiments

4.1 Experimental Settings

Datasets This paper evaluates the CTM-Explainer using four real-world temporal graph datasets: **Wikipedia**, which details user collaborations through edit histories, **Reddit**, capturing user interactions within subreddits, and **Enron**, which records email communications within the Enron corporation. Besides, two synthetic datasets [15], named **Synthetic v1** and **Synthetic v2**, are applied.

TGNNs and Baselines This study conducts a rigorous counterfactual analysis to comparatively evaluate CTM-Explainer against state-of-the-art baselines, including **GNN-Explainer** [42], **PG-Explainer** [43] and **TGNNExplainer** [15]. Utilizing a consistent strategy for explanation generation, we assess the impact of removing identified motifs on the predictions of **TGAT** [24] and **TGN** [23].

Implementation Details In this paper, we utilize the Pytorch framework to support the experiments with a workstation equipped with an NVIDIA GeForce RTX 3090, 64GB memory, and an 11th Gen Intel(R) Core(TM) i9-11900K @ 3.50GHz CPU. For the CTM-Explainer, two-layer MLP with a hidden layer dimension of 128 is utilized in policy networks. The Adam optimizer is selected to facilitate the learning process. To ensure effective training, a learning rate scheduler named ReduceLROnPlateau is adopted, with an initial learning rate set to 0.1, a reduction factor of 0.5, and a patience parameter of 2. During the training of the CTM-Explainer, 500 samples are randomly chosen for the training set across three explanation models. For Wikipedia, Reddit, Synthetic v1, and Synthetic v2, the training epochs are set to 16, 10, 10, and 10 respectively. For each dataset, we randomly select a test set of 800 samples, and for each test sample, 20 candidate events for explanation are identified. To account for variability in the experiments, results are compared across sparsity levels ranging from 0.05 to 0.95.

Table 1 Best CFF and AUFCFSC of Explanatory Models for TGN

Explainer	Wikipedia		Reddit		Synthetic v1		Synthetic v2		Enron	
	Best CFF↓	AUFCFSC↓	Best CFF↓	AUFCFSC↓	Best CFF↓	AUFCFSC↓	Best CFF↓	AUFCFSC↓	Best CFF↓	AUF-CFSC↓
GNN-Explainer	−2.6007	−0.9796	−3.1521	−0.6228	−13.9246	−6.2003	−21.6396	−10.3808	−2.1695	−0.7415
PG-Explainer	−2.5571	−1.0307	−3.2243	−0.6718	−12.8462	−6.7191	−24.0869	−16.2969	−2.2406	−0.8201
TGNNExplainer	−2.829	−1.1681	−3.4706	−0.7493	**−14.9658**	**−7.764**	**−26.0653**	−15.9241	−2.0008	−0.7758
CTM-Explainer	**−3.0729**	**−1.2181**	**−5.0931**	**−0.8505**	−11.4863	−6.7645	−24.6409	**−16.5049**	**−2.6934**	**−1.0106**

Table 2 Best CFF and AUFCFSC of Explanatory Models for TGAT

Explainer	Wikipedia		Reddit		Synthetic v1		Synthetic v2		Enron	
	Best CFF↓	AUFCFSC↓	Best CFF↓	AUFCFSC↓	Best CFF↓	AUFCFSC↓	Best CFF↓	AUFCFSC↓	Best CFF↓	AUF-CFSC↓
GNN-Explainer	−4.8081	−1.5889	−6.5111	−1.6975	−8.5877	−3.4522	−6.5206	−3.0095	−1.2634	−0.3607
PG-Explainer	−5.0027	−1.9435	−6.8813	−2.0158	−9.7671	−6.8187	−6.3332	−1.9237	−1.1021	−0.3791
TGNNExplainer	−5.3947	−2.245	−6.9408	−2.0021	−10.135	−6.843	−7.2003	−2.7452	−1.4773	**−0.4895**
CTM-Explainer	**−5.7106**	**−1.9972**	**−8.4207**	**−2.0472**	**−10.1353**	**−7.2416**	**−8.4668**	**−5.4730**	**−1.5061**	−0.4168

The random seed is set to 66. While CTM-Explainer supports synthetic event addition, all quantitative comparisons use event removal to align with baseline methods.

4.2 Evaluation Metrics

In counterfactual explanations, the goal is to find the smallest changes that alter outcomes. We redefine fidelity [15] as Counterfactual Fidelity (CFF) to measure the weakened impact of an explanation on the TGNN's prediction when CTM are removed. Additionally, we employ sparsity and the Area Under the Counterfactual Fidelity-Sparsity Curve (AUFCFSC) as metrics to evaluate the conciseness and overall quality of the explanations.

CFF: CFF is computed by the difference between the original prediction $f(\mathcal{G}(t))$ and the prediction after the counterfactual CTM have been removed $f(\mathcal{G}_{\mathrm{ctm}}(t))$, formulated as:

$$\begin{aligned}&\mathrm{CFF}(f(\mathcal{G}(t)), f(\mathcal{G}_{\mathrm{ctm}}(t)))\\&\quad= 1(y=1)(f(\mathcal{G}(t)) - f(\mathcal{G}_{\mathrm{ctm}}(t)))\\&\qquad+ 1(y=0)(f(\mathcal{G}_{\mathrm{ctm}}(t)) - f(\mathcal{G}(t))),\end{aligned} \tag{19}$$

where $\mathcal{G}_{\mathrm{ctm}}(t)$ denotes the graph with the CTM removed, and 1 denotes the indicator function. The 1 helps to adjust that a lower CFF signifies a more necessary counterfactual explanation, indicating the removed CTM is pivotal for the prediction. Moreover, we utilize the **Best CFF**, which measures the minimum achievable CFF, reflecting the best counterfactual explanation.

Sparsity: Sparsity represents the ratio of the number of events in the explanation $|E_{\mathrm{ctm}(t)}|$ to the total number of candidate events $|C|$, indicating the explanation's conciseness:

$$\mathrm{Sp} = \frac{|E_{\mathrm{ctm}}(t)|}{|C|}. \tag{20}$$

AUFCFSC: This metric synthesizes trade-offs between CFF and Sparsity across sparsity levels, offering a unified measure of counterfactual explanation quality.

4.3 Performance Comparison

To rigorously validate the effectiveness of CTM-Explainer, we compared it with GNN-Explainer, PG-Explainer, and T-GNNExplainer on five datasets: Wikipedia, Reddit, Enron, Synthetic v1, and Synthetic v2. The evaluation considered both TGN and TGAT as backbone models. Metrics included Best CFF, which reflects the strongest counterfactual explanatory ability, and AUFCFSC, which measures the overall trade-off between fidelity and sparsity.

The results in Tables 1 and 2 show that CTM-Explainer achieves the best overall performance in both backbones. On TGN–Reddit, CTM-Explainer reduces the Best CFF to −5.0931, compared with −3.2243 for PG-Explainer and −3.4706 for T-GNNExplainer. On TGN–Enron, it reaches −2.6934, clearly lower than −2.0008 from T-GNNExplainer. Under TGAT, the improvements are even more evident: on Wikipedia the Best CFF reaches −5.7106, and on Synthetic v2 the score is −8.4668, both outperforming all baselines. AUFCFSC values also confirm this advantage, since CTM-Explainer consistently yields the smallest areas, showing that fidelity remains stable even when explanations are constrained to be sparse. These results highlight that CTM-Explainer secures leading performance not only on average but also across diverse datasets with distinct temporal characteristics.

Figure 3 further illustrates the performance of explanatory models across different sparsity levels. CTM-Explainer maintains superior fidelity in most cases, demonstrating its ability to adapt to varying explanation sizes and practical deployment needs. On synthetic datasets, the advantage narrows as sparsity increases, and in some settings T-GNNExplainer performs competitively. This suggests that heuristic exploration can still provide benefits when temporal structures are simplified or artificially constrained. Nevertheless, CTM-Explainer's consistent dominance in real-world datasets confirms that causality-guided counterfactual reasoning offers a more reliable and generalizable solution.

Taken together, the comparative results emphasize both the strengths and limitations of CTM-Explainer. Its integration of ICE with reinforcement learning enables precise and compact explanations across TGN and TGAT, while minor performance gaps on synthetic graphs point to opportunities for hybrid strategies that combine causal reasoning with heuristic exploration. Overall, CTM-Explainer demonstrates a robust explanatory advantage and strong potential for practical deployment in temporally dynamic and causally complex scenarios.

4.4 Ablation Study

To validate the necessity of each design choice in CTM-Explainer, we construct three ablation variants. CTM-Random adopts the same framework as CTM-Explainer but skips training, using only randomly initialized parameters to guide event selection. This variant tests whether explanation quality arises from principled optimization or merely from the framework structure itself. CTM-GIB replaces the ICE-based causal objective with the Graph Information Bottleneck, a widely used information-theoretic principle, to examine whether correlation-based compression can substitute for causal reasoning. CTM-w/o T removes temporal embeddings from event representations, leaving only structural and attribute features, in order to test the necessity of explicit temporal encoding.

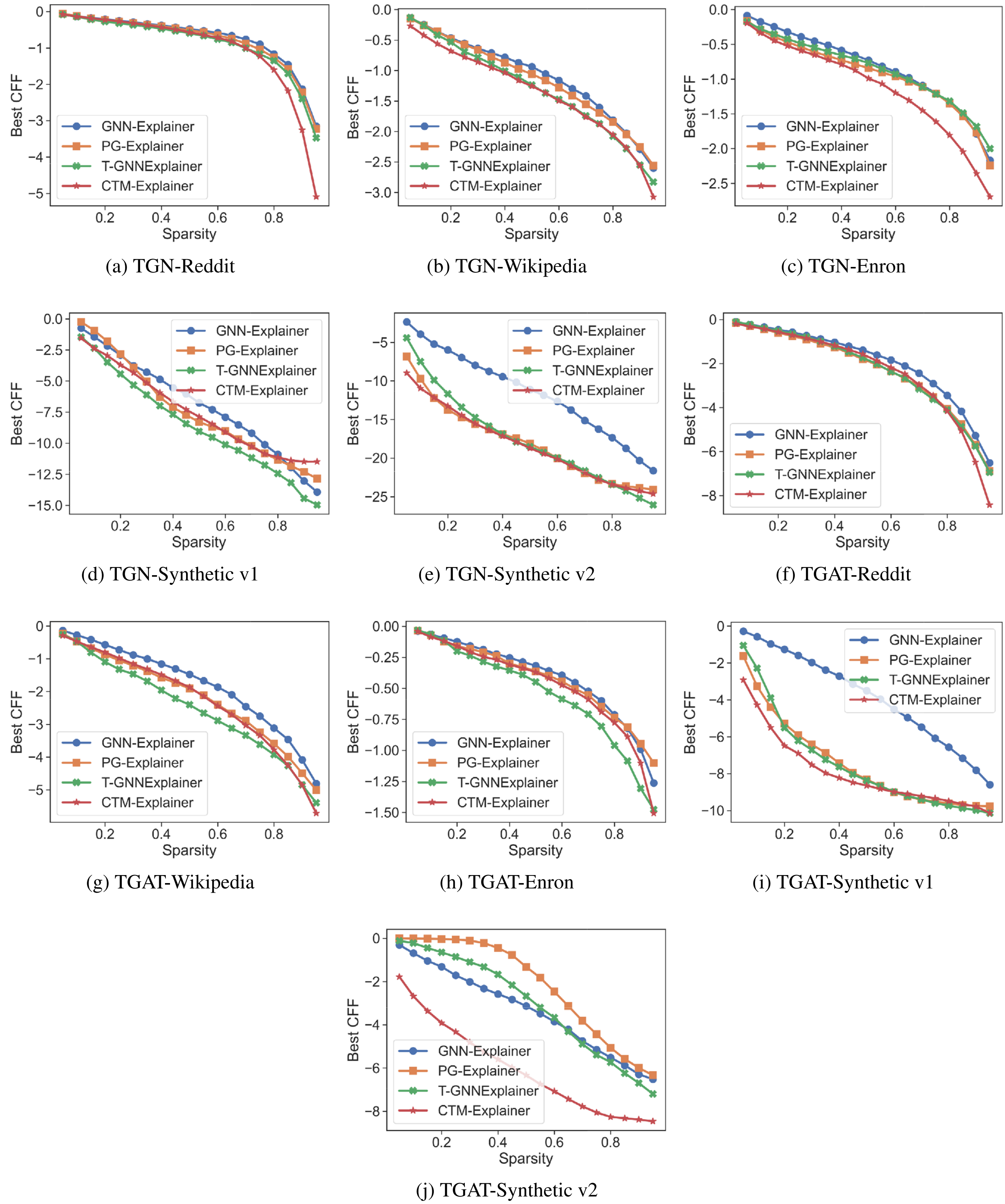
The results in Table 3 provide several insights into the contribution of each component. CTM-Random yields the weakest performance across all datasets, demonstrating that explanation quality does not come from the framework alone but requires a trained optimization policy. Moving to CTM-GIB, we observe that replacing the causal ICE objective with an information-theoretic alternative leads to mixed results: in some cases, such as TGN–Synthetic v1, it achieves relatively strong Best CFF, yet its AUFCFSC scores remain consistently higher. This indicates that correlation-based compression may capture partial relevance but fails to balance sparsity and fidelity in a stable manner. Finally, CTM-w/o T achieves results that are sometimes close to, or even slightly better than, the full model (e.g., TGN–Enron with Best CFF = −2.8656). This outcome is not contradictory but rather reflects the effect of our candidate construction strategy, which already confines events within k hops and τ timestamps of the target. Such strict spatio-temporal constraints preserve local temporal order while filtering out distant or irrelevant interactions. As a result, explicit temporal embeddings provide less additional gain in this restricted setting, since much of the essential temporal information is implicitly encoded by the candidate selection process. This does not imply that temporal modeling is unnecessary; on the contrary, temporal embeddings remain crucial in general scenarios without such localization, but their marginal contribution becomes less visible under our constrained candidate construction. In this way, the global temporal graph is effectively localized into a bounded, approximately acyclic subgraph, thereby reducing confounding and making the do-calculus assumptions reasonable. These findings confirm that the optimization policy, causal objective, and temporal encoding are all indispensable, thereby demonstrating that the overall design of CTM-Explainer is both theoretically justified and empirically validated.

4.5 Time-efficiency Comparison

To assess computational efficiency, we compare CTM-Explainer with T-GNNExplainer, a baseline specifically designed for TGNNs. As shown in Table 4, CTM-Explainer achieves one to two orders of magnitude lower computation time across all datasets. For example, on TGN–Reddit, T-GNNExplainer requires over 31,000 seconds, whereas CTM-Explainer completes 800 test samples in under 500 seconds. This advantage arises because T-GNNExplainer relies on Monte Carlo Tree Search, which repeatedly expands candidate subsets at high cost, while CTM-Explainer adopts a reinforcement learning strategy guided by causal effects that incrementally selects events without exhaustive exploration. These results demonstrate that CTM-Explainer not only maintains explanatory quality but also scales efficiently, rendering it well-suited for large-scale temporal graph applications.

4.6 Case Study

To scrutinize the divergent explanatory outputs, we perform a targeted case study of the PG-Explainer, T-GNNExplainer, and CTM-Explainer on the TGAT-Reddit and TGN-Reddit experiments, focusing on the pivotal event with the identifier 468158 under three distinct sparsity constraints: 0.2, 0.4, and 0.6 in Figures 4 and 5. The assigned weights to events, as generated by the explanatory models, are conveyed through bar charts. For the CTM-Explainer, the weights are



(a) TGN-Reddit (b) TGN-Wikipedia (c) TGN-Enron

(d) TGN-Synthetic v1 (e) TGN-Synthetic v2 (f) TGAT-Reddit

(g) TGAT-Wikipedia (h) TGAT-Enron (i) TGAT-Synthetic v1

(j) TGAT-Synthetic v2

Fig. 3 Performance of Explanatory Models across Different Sparsity Constraints

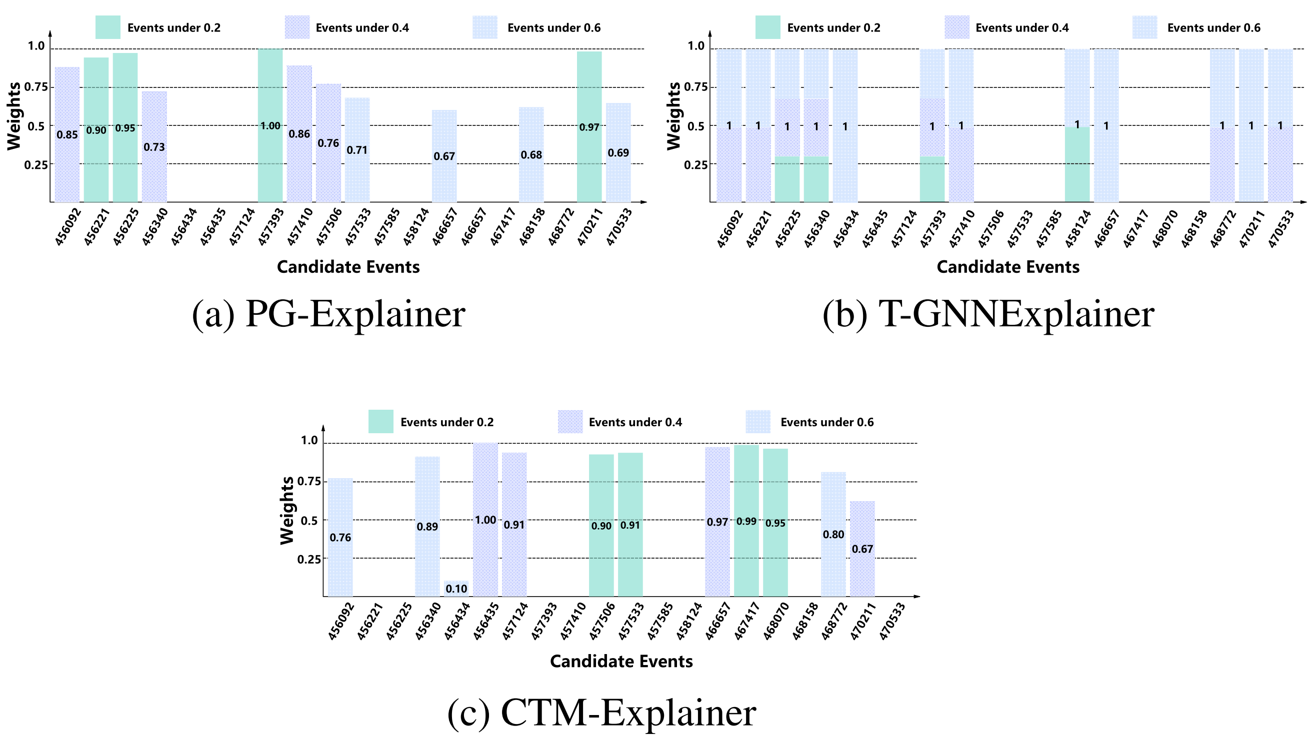
Table 3 Ablation Study for CTM-Explainer

Model	Dataset	Best CFF↓				AUFCFSC↓			
		CTM-Random	CTM-GIB	CTM-w/o T	CTM-Explainer	CTM-Random	CTM-GIB	CTM-w/o T	CTM-Explainer
TGAT	Wikipedia	−5.669	−5.6523	−5.6748	**−5.7106**	−1.8814	−1.7239	−1.6643	**−1.9972**
	Reddit	−8.3896	−8.4053	−8.4109	**−8.4207**	−2.1118	−2.0537	**−2.1146**	−2.0472
	Synthetic v1	−9.7503	−10.0979	−8.9983	**−10.1353**	−6.7385	−6.0883	−7.0772	**−7.2416**
	Synthetic v2	−6.3725	−7.4041	−7.7581	**−8.4668**	−2.917	−4.8899	−5.2906	**−5.473**
	Enron	−1.4521	−1.4923	−1.4829	**−1.5061**	−0.4204	**−0.4423**	−0.322	−0.4168
TGN	Wikipedia	−3.0491	−3.0711	−3.0593	**−3.0729**	−1.1786	−1.198	−1.0372	**−1.2181**
	Reddit	−5.069	−5.0638	−5.0902	**−5.0931**	−0.8147	−0.8439	**−0.8944**	−0.8505
	Synthetic v1	−10.5874	**−11.5828**	−9.7123	−11.4863	−5.613	−5.7055	−4.9974	**−6.7645**
	Synthetic v2	−24.0668	−22.9647	−23.7719	**−24.6409**	−15.8415	−10.2592	**−16.5721**	−16.5049
	Enron	−2.822	−2.842	**−2.8656**	−2.8276	−0.8385	−0.8778	−0.7955	**−0.8954**

Table 4 Computation Time Comparison (unit: seconds)

Experimental Groups	T-GNNExplainer	CTM-Explainer
TGN-Wikipedia	17002.44	112.13
TGN-Reddit	31874.37	492.28
TGN-Enron	17225.65	414.41
TGN-Synthetic v1	10455.74	104.30
TGN-Synthetic v2	11042.14	115.35
TGAT-Wikipedia	20736.65	155.18
TGAT-Reddit	37540.40	535.83
TGAT-Enron	27787.30	465.28
TGAT-Synthetic v1	18106.60	160.59
TGAT-Synthetic v2	18615.20	170.30

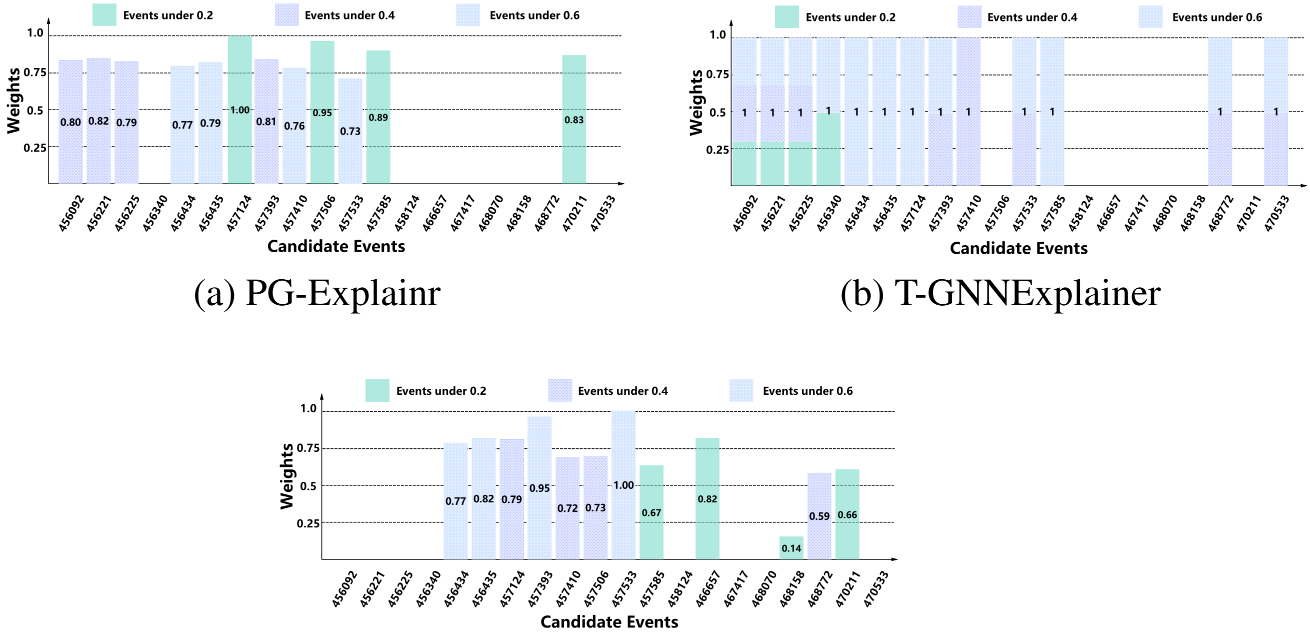
delineated by the ICE values, which reflect the influence of each event on both the previous preceding CTM and the TGNN's prediction. In contrast, the T-GNNExplainer, lacking an intrinsic scoring system, uniformly assigns a weight of 1 to all events within the purview of its explanations. The visualizations reveal that when explaining predictions for the same sample, the events deemed important by the explanatory models vary under a sparsity constraint of 0.2. However, as the constraint increases, there is a notable convergence in the set of explanations. This suggests that the variability in the explanatory models is particularly evident when generating more concise sets of explanations. Furthermore, a comparison between Figures 4 and 5 indicates



(a) PG-Explainer (b) T-GNNExplainer

(c) CTM-Explainer

Fig. 4 Case Study of Explanatory Models for explaining TGAT's outcome with Event 468158 in Reddit



(a) PG-Explainr (b) T-GNNExplainer

(c) CTM-Explainer

Fig. 5 Case Study of Explanatory Models for explaining TGN's outcome with Event 468158 in Reddit

that the explanatory outcomes vary across models even with identical input data, underscoring the explanatory models' ability to capture intrinsic characteristics of the TGNN's decision-making process.

5 Conclusion

In conclusion, this paper introduces the CTM-Explainer, a novel counterfactual explanation approach that significantly enhances the explainability of TGNNs. By generating CTM through a sequential process, the CTM-Explainer identifies the influence in the graph's temporal events that can lead to different TGNN's outcomes, providing clear and actionable insights into the TGNN's decision-making process. The integration of ICE analysis with a reinforcement learning framework ensures that the generated CTM are causally significant, offering a more targeted and efficient explanatory model. The extensive experiments on both real-world and synthetic datasets demonstrate the CTM-Explainer's ability. In future work, we plan to enhance the robustness and stability of the explanatory model under noisy or adversarial temporal dynamics, while further tailoring CTM-Explainer to domain-specific tasks such as clinical event forecasting and financial fraud detection, thereby improving both explainability and task performance in high-stakes environments.

Author Contributions All authors contributed to the study conception and design. Yibowen Zhao and Liu Ning conceive the study and develop the theoretical framework and inference methodology. Yibowen Zhao collects the data and implements the software. Yibowen Zhao and Yonghui Xu write the first draft of the manuscript. All authors critically revise and edit the manuscript. Lizhen Cui and Qingzhong Li supervise the project, provide strategic guidance, and secure funding. All authors read and approve the final manuscript.

Funding This research is partially supported by the Natural Science Foundation of China (No. 92367202, No. 62202279), the Excellent Youth Science Fund Project (Overseas) of Shandong Province (No. 2023HWYQ-039), and the Natural Science Foundation of Shandong Province (No. ZR2022QF018, No. ZR2023LZH006).

Data Availability All datasets used in this study are publicly available. The complete source code and any additional materials will be released via a public repository upon acceptance of this manuscript.

Declarations

Ethics approval and consent to participate Not applicable.

Consent for publication Not applicable.

Competing interests The authors declare that they have no competing interests.

References

1. Huang S, Poursafaei F, Danovitch J, Fey M, Hu W, Rossi E, Leskovec J, Bronstein M, Rabusseau G, Rabbany R (2024) Temporal graph benchmark for machine learning on temporal graphs. Advances in Neural Information Processing Systems **36**
2. Sun C, Ning Y, Shen D, Nie T (2024) Graph neural network-based short-term load forecasting with temporal convolution. Data Sci Eng 9(2):113–132
3. Jung J, Jung J, Kang U (2021) Learning to walk across time for interpretable temporal knowledge graph completion. In: Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining, pp. 786– 795
4. Yao Y, Joe-Wong C (2021) Interpretable clustering on dynamic graphs with recurrent graph neural networks. Proceedings of the AAAI Conference on Artificial Intelligence 35:4608–4616
5. Han Z, Chen P, Ma Y, Tresp V (2020) Explainable subgraph reasoning for forecasting on temporal knowledge graphs. In: International Conference on Learning Representations
6. Liu Y, Wang L, Yang P, Zhou L (2024) Mining interpretable regional co-location patterns based on urban functional region division. Data Sci Eng. https://doi.org/10.1007/s41019-024-00256-9
7. Zhu M, Shen D, Xu L, Wang X (2021) Scalable multi-grained cross-modal similarity query with interpretability. Data Sci Eng 6(3):280–293
8. Li L, Duan L, Wang J, He C, Chen Z, Xie G, Deng S, Luo Z (2023) Memory-enhanced transformer for representation learning on temporal heterogeneous graphs. Data Sci Eng 8(2):98–111
9. Seo S, Kim S, Jung J, Lee Y, Park C (2024) Self-explainable temporal graph networks based on graph information bottleneck. In: Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, pp. 2572– 2583
10. Verdone A, Scardapane S, Panella M (2024) Explainable spatio-temporal graph neural networks for multi-site photovoltaic energy production. Appl Energy 353:122151
11. Wu T, Ren H, Li P, Leskovec J (2020) Graph information bottleneck. Adv Neural Inf Process Syst 33:20437–20448
12. Zheng K, Yu S, Li B, Jenssen R, Chen B (2024) Brainib: Interpretable brain network-based psychiatric diagnosis with graph information bottleneck. IEEE Transactions on Neural Networks and Learning Systems
13. Fang J, Li X, Sui Y, Gao Y, Zhang G, Wang K, Wang X, He X (2024) Exgc: bridging efficiency and explainability in graph condensation. Proceedings of the ACM on Web Conference 2024:721–732
14. Chen J, Ying R (2024) Tempme: Towards the explainability of temporal graph neural networks via motif discovery. Advances in Neural Information Processing Systems **36**
15. Xia W, Lai M, Shan C, Zhang Y, Dai X, Li X, Li D (2023) Explaining temporal graph models through an explorer-navigator framework. In: The Eleventh International Conference on Learning Representations
16. O'Shaughnessy M, Canal G, Connor M, Rozell C, Davenport M (2020) Generative causal explanations of black-box classifiers. Adv Neural Inf Process Syst 33:5453–5467
17. Zhao K, Zhang L (2024) Causality-inspired spatial-temporal explanations for dynamic graph neural networks. In: The Twelfth International Conference on Learning Representations
18. Miraki A, Dapkutė A, Šiožinys V, Jonaitis M, Arghandeh R (2023) Causal-based spatio-temporal graph neural networks for industrial internet of things multivariate time series forecasting. In: World Conference on Explainable Artificial Intelligence, pp. 120– 130 . Springer
19. Heskes T, Sijben E, Bucur IG, Claassen T (2020) Causal shapley values: exploiting causal knowledge to explain individual predictions of complex models. Adv Neural Inf Process Syst 33:4778–4789
20. Choi s, park h, hwang s-w (2020) Meta-supervision for attention using counterfactual estimation. Data Sci Eng 5:193–204
21. Chen W, Cai R, Hao Z, Yuan C, Xie F (2020) Mining hidden non-redundant causal relationships in online social networks. Neural Comput Appl 32:6913–6923
22. Frye C, Rowat C, Feige I (2020) Asymmetric shapley values: incorporating causal knowledge into model-agnostic explainability. Adv Neural Inf Process Syst 33:1229–1239
23. Rossi E, Chamberlain B, Frasca F, Eynard D, Monti F, Bronstein MM (2020) Temporal graph networks for deep learning on dynamic graphs. International Conference on Machine Learning
24. Xu D, Ruan C, Körpeoglu E, Kumar S, Achan K (2020) Inductive representation learning on temporal graphs. In: 8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020
25. Ying Z, Bourgeois D, You J, Zitnik M, Leskovec, J (2019) Gnnexplainer: Generating explanations for graph neural networks. Advances in neural information processing systems **32**
26. Luo D, Cheng W, Xu D, Yu W, Zong B, Chen H, Zhang X (2020) Parameterized explainer for graph neural network. Adv Neural Inf Process Syst 33:19620–19631
27. Vu M, Thai MT (2020) Pgm-explainer: probabilistic graphical model explanations for graph neural networks. Adv Neural Inf Process Syst 33:12225–12235
28. Wu Z, Wang J, Du H, Jiang D, Kang Y, Li D, Pan P, Deng Y, Cao D, Hsieh C-Y et al (2023) Chemistry-intuitive explanation of graph neural networks for molecular property prediction with substructure masking. Nat Commun 14(1):2585
29. Chen X, Cai R, Zhu Y, Horwood J, Hao Z, Li Z, Hernández-Lobato JM, et al (2024) Feature attribution with necessity and sufficiency via dual-stage perturbation test for causal explanation. In: Forty-first International Conference on Machine Learning
30. Ye Z, Huang R, Wu Q, Liu Q (2024) Same: Uncovering gnn black box with structure-aware shapley-based multipiece explanations. Advances in Neural Information Processing Systems **36**
31. Akkas S, Azad A (2024) Gnnshap: Scalable and accurate gnn explanation using shapley values. In: Proceedings of the ACM on Web Conference 2024, pp. 827– 838
32. Abrate C, Bonchi F (2021) Counterfactual graphs for explainable classification of brain networks. In: Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining, pp. 2495– 2504
33. Chhablani C, Jain S, Channesh A, Kash IA, Medya S (2024) Game-theoretic counterfactual explanation for graph neural networks. In: Proceedings of the ACM on Web Conference 2024, pp. 503– 514
34. Qiu D, Wang M, Khan A, Wu Y (2024) Generating robust counterfactual witnesses for graph neural networks. In: 2024 IEEE 40th International Conference on Data Engineering (ICDE), pp. 3351– 3363 . IEEE Computer Society
35. Bajaj M, Chu L, Xue ZY, Pei J, Wang L, Lam PC-H, Zhang Y (2021) Robust counterfactual explanations on graph neural networks. Adv Neural Inf Process Syst 34:5644–5655

36. Kang H, Han G, Park H (2024) Unr-explainer: Counterfactual explanations for unsupervised node representation learning models. In: The Twelfth International Conference on Learning Representations (2024)
37. Verma S, Armgaan B, Medya S, Ranu S (2024)Induce: Inductive counterfactual explanations for graph neural networks. Transactions on Machine Learning Research
38. Li J, Zhang C, Zhang C (2023) Heterogeneous temporal graph neural network explainer. In: Proceedings of the 32nd ACM International Conference on Information and Knowledge Management
39. Lin W, Lan H, Li B (2021) Generative causal explanations for graph neural networks. In: International Conference on Machine Learning, pp. 6666– 6679 . PMLR
40. Sui Y, Wang X, Wu J, Lin M, He X, Chua T-S (2022) Causal attention for interpretable and generalizable graph classification. In: Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, pp. 1696– 1705
41. Wang X, Wu Y, Zhang A, Feng F, He X, Chua T-S (2022) Reinforced causal explainer for graph neural networks. IEEE Transactions on Pattern Analysis and Machine Intelligence
42. Ying Z, Bourgeois D, You J, Zitnik M, Leskovec J (2019) Gnnexplainer: Generating explanations for graph neural networks. Advances in neural information processing systems **32**
43. Luo D, Cheng W, Xu D, Yu W, Zong B, Chen H, Zhang X (2020) Parameterized explainer for graph neural network. Advances in neural information processing systems