A Topological Encoder Decoder Framework for Temporal Graph Learning

Ronan Buck
University of Central Florida
Orlando, FL, USA

Kiarash Shamsi
University of Manitoba
Winnipeg, Canada

Tran Gia Bao Ngo
University of Manitoba
Winnipeg, Canada

Astrit Tola
Florida State University
Tallahassee, FL, USA

Baris Coskunuzer
The University of Texas at Dallas
Richardson, TX, USA

Cuneyt Gurcan Akcora
University of Central Florida
Orlando, FL, USA

Abstract

Most temporal graph learning methods reduce future prediction to discriminative inference over past interactions, focusing on edges or node labels for a largely persistent set of nodes. This view breaks down when new nodes and edges appear, since the future graph becomes a new object with changing size, composition, and structure. We address this gap by framing temporal graph prediction as an *inverse topology* problem. Instead of predicting edges directly, we first predict a multiscale topological descriptor of the future graph and then reconstruct a plausible future snapshot that realizes this descriptor under inductive constraints. This approach makes global structure and node churn explicit prediction targets and produces forecasted graphs that downstream tasks can evaluate without retraining. Across 14 temporal graph datasets, we evaluate our method, TopoGED, on node, link, and graph property prediction and compare it against state-of-the-art temporal graph models. TopoGED significantly improves node-level forecasting accuracy over the strongest baseline, with a macro average of 0.49 versus 0.05. It also outperforms baselines in 60% of graph-structure metric evaluations and substantially increases macro-average edge prediction metrics, from near-zero to 0.12. Our results show that topology-guided graph forecasting can predict inductive future snapshots whose structure supports multiple downstream evaluations.

1 Introduction

Dynamic graphs, in which nodes and edges evolve over time, are central to modeling real systems such as social networks, citation graphs, online communities, and blockchain transaction networks [50, 8]. In these settings, the object of interest is often not a single future edge or label, but the next network state itself. A risk signal for an ERC20 token[52], a stress indicator for a transaction ecosystem, or a moderation statistic for an online forum depends on the structure of the future graph: which nodes remain active, which new nodes arrive, and how connectivity reorganizes.

Most temporal graph learning methods approach this problem through discriminative prediction. They learn temporal representations from past interactions and use them to score candidate edges or node labels [45, 42, 31]. This paradigm is effective when the future graph is assumed to be an extension of the past over a largely persistent node set. However, this assumption breaks down under node churn. In many real temporal graphs, new nodes appear, old nodes disappear and reappear, and edges form between both existing and newly arriving nodes. In this regime, future prediction is no longer only a ranking problem over known node pairs; it becomes a graph reconstruction problem with changing size, composition, and topology.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

We argue that this setting is more naturally addressed through *inductive snapshot forecasting*. Given a sequence of graph snapshots up to time t, the goal is to predict the node set and structure for the next snapshot $\widehat{\mathcal{G}}_{t+1}$. This forecast-then-predict view aligns with deployment: downstream decisions are made before the next graph is observed, and many downstream quantities can be computed only after constructing a future graph. Thus, a temporal graph model should not only rank candidate edges, but also produce a coherent future graph on which new tasks can be evaluated.

Topology [4] is useful in this setting because it compactly describes graph structure across scales while abstracting away exact node identities. Two snapshots may differ in size and node labels, yet still share similar structural organization, such as heavy-tailed degree patterns, hub-centered motifs, or changes in edge density. Rather than predicting edges directly, we summarize each snapshot by a multiscale filtration descriptor that records how node and edge mass accumulate across filtration thresholds [4]. This descriptor captures the graph's structural state, including its final node and edge budgets, while remaining lightweight enough to forecast over time.

This leads to our central formulation: temporal graph forecasting as an *inverse topology problem*. Instead of predicting individual edges, we first predict the future topological descriptor $\widehat{\Phi}(\mathcal{G}_{t+1})$. We then reconstruct a plausible future graph whose structure realizes this predicted descriptor while respecting inductive constraints such as node arrivals, node reappearances, and edge types involving new nodes. Because the descriptor is not injective, many non-isomorphic graphs may share the same topological signature. Our decoder resolves this ambiguity by combining global topological constraints with learned edge-formation scores, seeking a reconstruction that has low descriptor error and matches empirical interaction patterns.

We introduce **TopoGED** (**Topo**logical **G**raph **E**ncoder **D**ecoder), a fully inductive encoder–decoder framework for discrete-time temporal graph forecasting. TopoGED encodes each snapshot through a degree-based filtration descriptor and forecasts the next descriptor together with node and edge budgets. A memory module estimates which historical vertices reappear; new unlabeled vertices are instantiated according to the predicted arrival budget, and a multi-phase decoder generates edges by inductive type: old–old recurring, old–old newly formed, old–new, and new–new. The resulting forecasts are complete future snapshots and can be used directly for downstream prediction without task-specific retraining. Our contributions are as follows:

- We formulate temporal graph learning under node churn as an inductive snapshot forecasting problem and define arrival-aware node and edge types, enabling explicit modeling of reappearing old nodes, newly arriving nodes, and edges involving new nodes.
- We introduce a topology-guided inverse formulation that forecasts a multiscale filtration descriptor and reconstructs a graph realization from the descriptor.
- We propose TopoGED, a topological encoder–decoder that combines descriptor prediction, memory-based node reappearance, and multi-phase edge decoding.
- We evaluate TopoGED across social, web and transaction networks, showing consistent gains in structural fidelity, node-level forecasting, edge-level prediction, and downstream graph-property prediction while reducing the time costs by 95% against the best competitor.

2 Related Work

Temporal graph models learn evolving node and edge representations, and are commonly modeled as discrete-time dynamic graphs (DTDGs) or continuous-time dynamic graphs (CTDGs). CTDGs methods [46, 19, 38, 25, 45, 23, 40, 42, 31] model the network as a stream of time-stamped events. DTDGs [34, 32, 37, 43] represent the network as a sequence of snapshot graphs collected at discrete timestamps. These approaches provide strong predictors, but most assume a fixed node universe or persistent node identities, limiting their ability to inductively generate unseen nodes and their interactions (Table 1). We focus on DTDGs because snapshot-level modeling offers a tractable framework for forecasting, enabling batch processing over time and explicit control of structural targets (e.g., node and edge budgets under churn). In contrast, CTDG generation relies on event-level modeling and point-process assumptions that are unnecessary for snapshot forecasting.

Dynamic graph generation. Dynamic graph generation has been studied in both discrete-time [48, 24] and continuous-time [9, 51, 10, 14] regimes. While effective for edge-event synthesis, they do

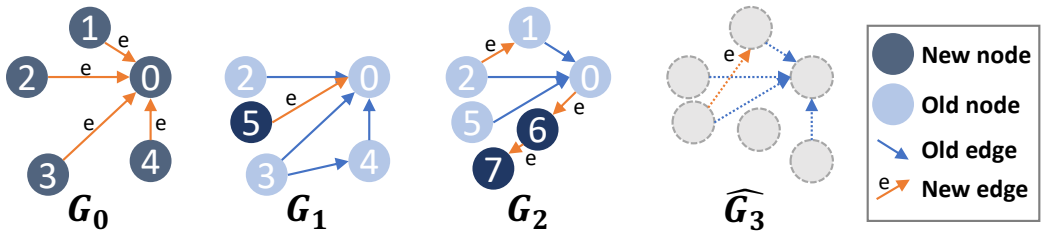


Figure 1: Temporal graph evolution and next-snapshot forecasting. Old nodes and edges have appeared in at least one earlier snapshot; new nodes and edges appear for the first time in the current snapshot. Topological encodings obtained through sublevel filtrations of the observed snapshots G_0, G_1, and G_2 are used to forecast $\widehat{G}_3$.

not directly yield coherent snapshot forecasts in the presence of churn (node disappearance) and are often tied to training node IDs. This leaves a gap for discrete-time snapshot-forecasting methods that can inductively predict future graphs with new nodes while preserving global structural fidelity.

Table 1: Temporal graph forecasting capabilities. o and n denote previously observed and newly arriving nodes; the n column denotes the forecast of new nodes (i.e., their count). Unseen $(o$–$o)$ denotes previously unseen edges between observed nodes. Full snapshot denotes joint forecasting of the future node and edge sets.

Method(s)	n	unseen $(o$–$o)$	$(o$–$n)$, $(n$–$n)$	Full snapshot	Objective
JODIE [23]	✗	✓	?	✗	IS
DyRep [40]	✗	✓	✓	✗	LT
TGAT, TGN, DyGPrompt [42, 31, 46]	✗	✓	✓	✗	LN
DySAT [32]	✗	✓	✓	✗	LP
EvolveGCN [28]	✗	✓	✓	✗	MP
HTGN [43]	✗	✓	✓	✗	LP
CTAN [9]	✗	✓	?	✗	LN
HOPE [25]	✗	✗	✗	✗	NT
SLATE [18]	✗	✓	✗	✗	LP
TIGGER-I, DG-Gen [10, 14]	✗	✓	✓	✗	GG
GraphPulse, MiNT [34, 35]	✗	✗	✗	✗	GP
TopoGED (ours)	✓	✓	✓	✓	SF

Topological and spectral methods. Topological and spectral methods incorporate structural priors into graph learning to capture patterns beyond local message passing, such as cycles and global connectivity changes. SLATE [18] uses temporal Laplacian features but does not address snapshot generation or inductive churn. Persistent homology supports graph-level readouts through differentiable vectorizations and layers [12, 3], while topology-aware GNNs use rewiring and topological augmentations [6, 13]. Related methods use topology to study structural evolution and regularize graph generation [44, 29]. Our method builds directly on TopER [39], which summarizes a graph using the slope and intercept of a linear fit between node and edge counts across filtration thresholds. We omit the line-fitting step but extend its filtration-based approach to temporal graph learning through compact multiscale signatures that support snapshot forecasting and generation.

3 Preliminaries

Dynamic graph concepts. We study discrete-time dynamic graphs represented as a sequence of snapshots $\mathcal{G} = \{\mathcal{G}_1, \ldots, \mathcal{G}_T\}$ with $\mathcal{G}_t = (\mathcal{V}_t, \mathcal{E}_t)$. We consider the inductive setting where the node set changes over time. For each node v, we define its first appearance time as $\tau(v) := \min\{t : v \in \mathcal{V}_t\}$. At time t, we partition the node set as $\mathcal{V}_t = \mathcal{V}_t^{\text{new}} \cup \mathcal{V}_t^{\text{old}}$, where $\mathcal{V}_t^{\text{new}} = \{v \in \mathcal{V}_t : \tau(v) = t\}$ and $\mathcal{V}_t^{\text{old}} = \{v \in \mathcal{V}_t : \tau(v) < t\}$. Nodes that disappear and later reappear are treated as *old* upon reappearance (see Figure 1). We decompose edges in $\mathcal{E}_t$ into four disjoint types based on the appearance of edge endpoints: old–old (oo), old–new (on), new–new (nn). Among old–old edges, we distinguish between previously observed pairs and newly formed ones. Following EdgeBank [30], we refer to these as *oo-bank* (*i.e.,* previously seen) and *oo-nobank* (*i.e.,* not seen) edges. Formally, we define:

$$\mathcal{E}_t^{\text{oo-bank}} = \{(u,v) \in \mathcal{E}_t : u, v \in \mathcal{V}_{\text{old}}(t), (u,v) \in \cup_{j=1}^{t-1} \mathcal{E}_j\}, \quad \mathcal{E}_t^{\text{on}} = \{(u,v) \in \mathcal{E}_t : u \in \mathcal{V}_{\text{old}}(t), v \in \mathcal{V}_{\text{new}}(t)\},$$
$$\mathcal{E}_t^{\text{oo-nobank}} = \{(u,v) \in \mathcal{E}_t : u, v \in \mathcal{V}_{\text{old}}(t), (u,v) \notin \cup_{j=1}^{t-1} \mathcal{E}_j\}, \quad \mathcal{E}_t^{\text{nn}} = \{(u,v) \in \mathcal{E}_t : u, v \in \mathcal{V}_{\text{new}}(t)\}.$$

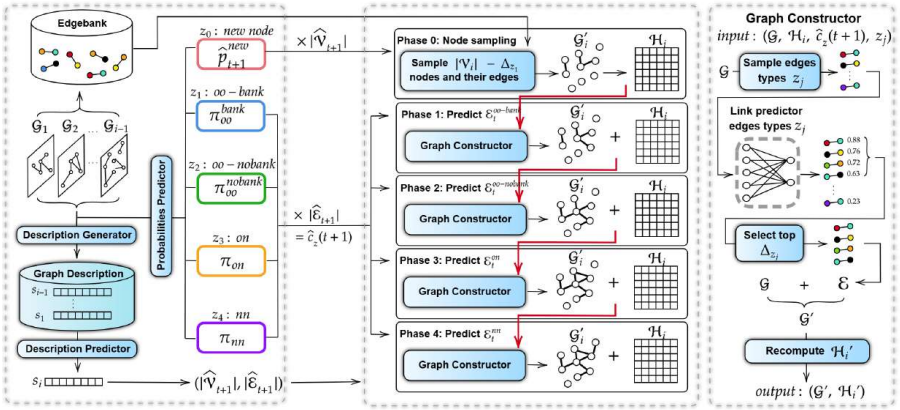


Figure 3: **TopoGED overview.** Each observed snapshot $\mathcal{G}_t$ is encoded into a multiscale topological descriptor $\Phi(\mathcal{G}_t)$ via filtration. A lightweight topology predictor forecasts the next-step descriptor $\Phi(\mathcal{G}_{t+1})$ together with coarse structural targets (node and edge budgets). Conditioned on these targets and a node-memory state, the phased decoder reconstructs the next snapshot $\widehat{\mathcal{G}}_{t+1}$.

Topological filtration. Topological data analysis [7] introduces the notion of a filtration to describe how the structure of a graph emerges across scales. By tracking the birth and merging of components and higher-order features as a scale parameter varies, filtrations provide a principled multiscale view of graph organization [29, 39]. However, computing full persistent homology on large or evolving graphs is often prohibitively expensive [1], both computationally and in memory, which limits its direct use in temporal graph learning. This motivates the use of lightweight filtration-based summaries that retain multiscale structural information while remaining scalable.

Given $\mathcal{G}$ and a filtration function, such as node degree, $f : \mathcal{V} \to \mathbb{R}$, an induced sublevel filtration using thresholds can be formed as $\epsilon_1 < \cdots < \epsilon_n$: $\mathcal{V}_i = \{v \in \mathcal{V} : f(v) \leq \epsilon_i\}, \mathcal{E}_i = \{(u, v) \in \mathcal{E} : u, v \in \mathcal{V}_i\}, \mathcal{G}_{\epsilon_1} \subseteq \cdots \subseteq \mathcal{G}_{\epsilon_n} = \mathcal{G}$. When the filtration level is clear, we omit the subscript i.

4 TopoGED: Topological Graph Encoder-Decoder Framework

Philosophy. We propose the view that the evolution of a graph can be understood through its multiscale structural shape, and that topology provides a principled way to make this shape comparable across time. In temporal graphs, snapshots may differ widely in size, node identity, and local connectivity due to churn, yet they often follow coherent structural patterns as they evolve. For example, in blockchain transaction networks [33], new nodes must transact with exchange nodes to buy coins, which creates star-shaped subgraphs around exchange nodes. Topological representations capture these patterns by summarizing how connectivity accumulates across scales, abstracting away from specific node labels and exact graph size. In this sense, topology can provide a structural blueprint across snapshots: it preserves the graph's organization while allowing scale- and node-specific effects to be learned downstream.

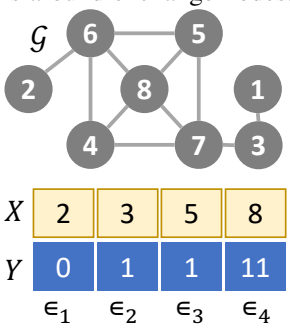


Figure 2: Filtering nodes/edges with 1 to 4 degree filtration. X and Y track the number of nodes/edges in a given filtration bucket.

Goal and high-level design. We consider discrete-time dynamic graphs with node churn. Given snapshots up to time t, TopoGED forecasts the next snapshot $\widehat{\mathcal{G}}_{t+1} = (\widehat{\mathcal{V}}_{t+1}, \widehat{\mathcal{E}}_{t+1})$ in an inductive setting where new nodes may appear. TopoGED consists of three components: (i) a scalable multiscale *topological descriptor* $\Phi(\mathcal{G}_t)$ for each snapshot, computed via TDA-style filtration sequences; (ii) a *topology predictor* that forecasts the next-snapshot descriptor $\Phi(\mathcal{G}_{t+1})$ together with coarse structural targets, including the edge budget $|\mathcal{E}_{t+1}|$, node budget $|\mathcal{V}_{t+1}|$, and the probability of new nodes p^{new}_{t+1}; and (iii) a *decoder* that reconstructs $\widehat{\mathcal{G}}_{t+1}$ by generating edges in four stages aligned with the four edge types defined in Section 3. Figure 3 provides an overview of our design.

4.1 Graph Encoder

For a snapshot $\mathcal{G}_t = (\mathcal{V}_t, \mathcal{E}_t)$, the encoder computes a filtration-based multiscale topological descriptor. Given the degree thresholds $\epsilon_1 < \cdots < \epsilon_n$ (by default $n = 10$), we form the sublevel filtration $\mathcal{G}_{t,\epsilon_1} \subseteq \cdots \subseteq \mathcal{G}_{t,\epsilon_n} = \mathcal{G}_t$. At each filtration level i, we record node and induced-edge counts, $x_{t,i} = |\mathcal{V}_{t,\epsilon_i}|$ and $y_{t,i} = |\mathcal{E}_{t,\epsilon_i}|$. This yields monotone sequences $X_t = (x_{t,1}, \ldots, x_{t,n})$ and $Y_t = (y_{t,1}, \ldots, y_{t,n})$. We denote the resulting topological descriptor of $\mathcal{G}_t$ by $\Phi(\mathcal{G}_t) \coloneqq (X_t, Y_t)$. By definition, at the final threshold ϵ_n, these sequences recover the snapshot sizes: $X_{t,n} = |\mathcal{V}_t|$ and $Y_{t,n} = |\mathcal{E}_t|$ (Fig. 2). See Appendix Figure 8 for example graphs and their descriptors.

We use the degree-based node filtration $f(v) = \deg(v)$, which is computationally inexpensive. Importantly, a degree provides a direct and efficient handle on the *inverse topology* problem: predicting a graph, given its topological description. Degree thresholds in the descriptor control node inclusion and induced edge counts, which we can use to construct a graph that realizes (*i.e.,* decodes) the predicted topological descriptor. For example, a star graph with four nodes has the descriptor $((3,0),(3,0),(4,3))$ for the degree thresholds $\epsilon_1, \epsilon_2, \epsilon_3$.

The following proposition supports using $\Phi(\mathcal{G}_{t+1})$ as a stable forecasting target by showing that, for a fixed node set, bounded edge noise induces bounded descriptor error; the churn model handles node-set changes separately. We defer the proof to Appendix D.

Proposition 4.1 (Stability under edge edits)**.** *Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ and $\mathcal{G}' = (\mathcal{V}, \mathcal{E}')$ be graphs on the same node set with $|\mathcal{E} \triangle \mathcal{E}'| \leq \delta$. Let X, Y and X', Y' be the degree-sublevel filtration node and induced-edge count sequences computed using thresholds $\epsilon_1 < \cdots < \epsilon_n$.*

(i) If the thresholds are consecutive degrees $\{0, 1, \ldots, D\}$, then $\|X - X'\|_1 \leq 2\delta$.

(ii) For any thresholds, $\|Y - Y'\|_1 \leq \delta \sum_{i=1}^{n} (1 + 2\epsilon_i)$.

4.2 Predictor for an Encoded Graph

Given the most recent k descriptors $\Phi(\mathcal{G}_{t-k+1}), \ldots, \Phi(\mathcal{G}_t)$, we use a lightweight temporal predictor to forecast the next-step descriptor $\widehat{\Phi}(\mathcal{G}_{t+1})$. Since the final entries of the descriptor recover snapshot size, this prediction also determines the node and edge budgets $|\widehat{\mathcal{V}}_{t+1}|$ and $|\widehat{\mathcal{E}}_{t+1}|$. The snapshot node count is learned from $|\hat{\mathcal{V}}_{t+1}|$, but decoding the next snapshot additionally requires anticipating node churn. We therefore jointly predict the fraction of newly appearing nodes at $t+1$, defined as $\hat{p}^{\text{new}}_{t+1} \coloneqq \frac{|\mathcal{V}^{\text{new}}_{t+1}|}{|\mathcal{V}_{t+1}|}$. Given the predicted node budget $|\widehat{\mathcal{V}}_{t+1}|$ and the predicted fraction of new nodes $\widehat{p}^{\text{new}}_{t+1}$, we determine the node counts $\widehat{n}^{\text{old}}_{t+1} = \lfloor |\widehat{\mathcal{V}}_{t+1}|(1 - \widehat{p}^{\text{new}}_{t+1}) \rfloor$ and $\widehat{n}^{\text{new}}_{t+1} = |\widehat{\mathcal{V}}_{t+1}| - \widehat{n}^{\text{old}}_{t+1}$. This separation isolates the reappearance problem from the arrival problem.

4.3 Memory Module for Node Reappearance

To identify $\widehat{n}^{\text{old}}_{t+1}$ reappearing nodes, we restrict candidates to the set of nodes observed in the last k snapshots memory window: $\mathcal{V}_{t-k:t} = \bigcup_{\tau=t-k}^{t} \mathcal{V}(\mathcal{G}_\tau)$. This window limits variance and reflects the empirical fact that reappearance likelihood decays with time. For each node $v \in \mathcal{V}_{t-k:t}$, we assign:

$$P(v,t) = \exp\left(-\frac{t - \ell_v}{\lambda}\right)\left(\frac{\deg_v}{\deg_{\max}}\right)^{\alpha} f_v^{\beta}, \tag{1}$$

where ℓ_v is the last observed time of v, $\deg_v$ is its degree aggregated over the window, $\deg_{\max}$ is the maximum degree in the window, and f_v is the number of snapshots in which v appears within the window. The exponential term enforces recency, the normalized degree captures structural prominence, and the frequency term captures persistence across snapshots. The normalization by $\deg_{\max}$ keeps scores on a comparable scale across datasets. We keep these factors separate to avoid conflating instantaneous connectivity with temporal activity.

We normalize $\{P(v,t)\}_{v \in \mathcal{V}_{t-k:t}}$ to a distribution and sample $\widehat{n}^{\text{old}}_{t+1}$ nodes without replacement. Sampling introduces diversity and prevents concentration on a small set of high-degree nodes, which improves stability under heavy-tailed degree distributions. We then introduce $|\widehat{\mathcal{V}}^{\text{new}}_{t+1}|$ nodes to complete $\widehat{\mathcal{V}}_{t+1}$. This step treats arrivals as a count prediction problem and avoids imposing identities where none exist.

4.4 Filtration-Guided Edge Budgets

TOPOGED reconstructs the predicted snapshot by generating edges in four stages aligned with the edge types in Section 3: old–old bank ($\mathcal{E}^{\text{oo-bank}}$), old–old nobank ($\mathcal{E}^{\text{oo-nobank}}$), old–new ($\mathcal{E}^{\text{on}}$), and new–new ($\mathcal{E}^{\text{nn}}$). Decoding is driven by three sources of information: (i) node churn targets produced by the predictor and memory module, (ii) edge-type budgets, and (iii) optional re-encoding between stages to update node representations as structure (through edges) accumulates.

Edge-type budgets. Let $|\widehat{\mathcal{E}}_{t+1}|$ be the predicted next snapshot edge count. We predict the edge-type proportions from their recent empirical history. For each snapshot, we compute $\boldsymbol{\pi}_t = (\pi_t^{\text{oo-bank}}, \pi_t^{\text{oo-nobank}}, \pi_t^{\text{on}}, \pi_t^{\text{nn}})$, where $\pi_t^z = |\mathcal{E}_t^z|/|\mathcal{E}_t|$ for each edge type $z \in \{\text{oo-bank}, \text{oo-nobank}, \text{on}, \text{nn}\}$. Given the last k proportion vectors, we forecast $\widehat{\boldsymbol{\pi}}_{t+1}$ with the same lightweight temporal predictor (e.g., a weighted average model such as V-EWMA[2]) used for the descriptor variables and renormalize it to lie in the simplex. The predicted total edge budget $|\widehat{\mathcal{E}}_{t+1}|$ is then divided across edge types as $\hat{c}_{t+1}^z = \lfloor \widehat{\pi}_{t+1}^z |\widehat{\mathcal{E}}_{t+1}| \rfloor$.

Filtration-guided budget guidance. In addition to the final node and edge budgets ($|\widehat{\mathcal{V}}_{t+1}|, |\widehat{\mathcal{E}}_{t+1}|$), the filtration sequences $\Phi(\mathcal{G}_t) := (X_t, Y_t)$. provide multiscale structural targets. We use these sequences to define intermediate checkpoints on cumulative node and edge counts after each decoding phase, and apply a soft penalty when the partially generated graph deviates from the predicted filtration profile. This guidance steers generation toward the forecasted multiscale structure without enforcing hard degree constraints at individual filtration levels. We elaborate on how the topological descriptor $\Phi(\mathcal{G}_t)$ guides graph construction step by step in Appendix C.1.

Phase 1: old–old bank edges. We begin by generating recurring edges between old nodes. The candidate set consists of previously seen edges whose endpoints are old nodes, $\{(u, v) \in \mathcal{E}_{t-i:t} : u, v \in \mathcal{V}_{t+1}^{\text{old}}\}$. A dedicated predictor head scores these pairs, and we add the highest-scoring pairs until we reach $\hat{c}_{t+1}^{\text{oo-bank}}$.

Phase 2: old–old nobank edges. Next, we generate previously unseen edges between old nodes. Candidate pairs $(u, v) \in \mathcal{V}_{t+1}^{\text{old}} \times \mathcal{V}_{t+1}^{\text{old}}$ are scored by a dedicated predictor head, and the highest-scoring pairs are added until reaching $\hat{c}_{t+1}^{\text{oo-nobank}}$.

Phase 3: old–new edges. We then connect new nodes to the existing graph. We initialize new nodes with cold-start embeddings (Section 4.5), then score and select old–new candidate pairs until we reach $\hat{c}_{t+1}^{\text{on}}$.

Phase 4: new–new edges. Finally, we generate edges among new nodes by scoring new–new candidate pairs and adding edges until we meet the budget $\hat{c}_{t+1}^{\text{nn}}$. This produces the completed snapshot $\widehat{\mathcal{G}}_{t+1}$. After generation, we update the node memory using $\widehat{\mathcal{G}}_{t+1}$.

4.5 Edge Decoder: Multi-Head Predictor

Edge generation is performed by a multi-head predictor with a shared embedding trunk. At each decoding phase, the trunk computes node embeddings $\{h_v\}_{v \in \widehat{\mathcal{V}}_{t+1}}$ on the current partially constructed graph using a GCN [21] over the generated adjacency. Four type-specific heads then score candidate pairs corresponding to the inductive edge categories:

$$M_{\text{oo-bank}} : (h_u, h_v) \mapsto p_{uv}^{(\text{oo-bank})}, \qquad M_{\text{oo-nobank}} : (h_u, h_v) \mapsto p_{uv}^{(\text{oo-nobank})},$$
$$M_{\text{on}} : (h_u, h_v) \mapsto p_{uv}^{(\text{on})}, \qquad M_{\text{nn}} : (h_u, h_v) \mapsto p_{uv}^{(\text{nn})}.$$

For edge type z, the corresponding head outputs probabilities $\hat{p}_{uv}^z \in [0, 1]$ for candidate pairs (u, v). Candidate pairs are ranked by $\hat{p}_{uv}^z$, and the top-ranked edges are added until the phase-specific budget $\widehat{c}_{t+1}^z$ is reached. To keep decoding tractable, each phase scores only a sampled candidate pool, whose size is chosen proportional to the predicted edge budget.

Cold start for new nodes. New nodes introduced in phases 3 and 4 initially have no incident edges and therefore lack structural context. We initialize their embeddings using degree-conditioned prototypes computed from the predicted old nodes. Let $\kappa(v)$ denote the degree bin assigned to a new node v from the predicted filtration and edge-budget targets.

For bin k, define $\mathcal{C}_k = \{u \in \widehat{\mathcal{V}}_{t+1}^{\text{old}} : \deg(u) = k\}$, and $h_v^{\text{new}} = \frac{1}{|\mathcal{C}_{\kappa(v)}|} \sum_{u \in \mathcal{C}_{\kappa(v)}} h_u$. When $\mathcal{C}_{\kappa(v)}$ is empty, we use the nearest non-empty bin, and fall back to a global prototype if no such bin exists. The binning scheme is detailed in Appendix C.3.

Since the filtration descriptor is not injective, many non-isomorphic graphs can realize the same predicted descriptor. The multi-phase decoder resolves this ambiguity by combining global filtration-guided targets with learned edge scores trained from historical positive and negative edges. This yields graph realizations that are encouraged to match the predicted multiscale structure while remaining consistent with empirical edge-formation patterns.

To the best of our knowledge, only the degree filtration lets us construct a graph from a given descriptor, but degree is discrete. **From the broader field of representation learning, we know that a smooth, non-discrete representation may yield better results. However, our long discussions could not find a smooth encoder; hence, we leave this as an open problem.**

4.6 Training and Inference

TopoGED is trained by supervising both the evolution of the topological descriptor and the edge generation, using two corresponding components. First, the topology predictor forecasts the next-step descriptor and probabilities. Given the past k encodings, it outputs $\widehat{\Phi}(\mathcal{G}_{t+1})$ and the predicted new-node fraction $\widehat{p}_{t+1}^{\text{new}}$, and is optimized with a regression loss against the true descriptor sequence.

Second, the edge decoder is trained with positive and negative edges from historical snapshots. For each inductive edge type $z \in \{\text{oo-bank}, \text{oo-nobank}, \text{on}, \text{nn}\}$, let $\mathcal{E}_{t+1}^z$ be the corresponding ground-truth edges in snapshot $t+1$. Each decoder head is trained using binary cross-entropy with negative sampling: $\mathcal{L}_{\text{edge}}^z = -\sum_{(u,v)\in\mathcal{E}_{t+1}^z} \log p_{uv}^z - \sum_{(u',v')\sim\mathcal{N}^z} \log\big(1 - p_{u'v'}^z\big)$, where $\mathcal{N}^z$ denotes uniformly sampled negative pairs from the corresponding candidate space. This encourages decoded graphs to match empirical edge-formation patterns. During training, we use *teacher forcing* [41]: each decoding phase conditions on the ground-truth partial graph formed by earlier-phase edges in $\mathcal{G}_{t+1}$, stabilizing optimization and decoupling phase-wise errors. The total decoder loss is $\sum_z \mathcal{L}_{\text{edge}}^z$, and the topology predictor and decoder are trained with separate optimizers.

Inference. At inference time, given snapshots up to time t and the current memory state, we predict $\widehat{\Phi}(\mathcal{G}_{t+1})$, the node and edge budgets $(|\widehat{\mathcal{V}}_{t+1}|, |\widehat{\mathcal{E}}_{t+1}|)$, and the new-node fraction $\widehat{p}_{t+1}^{\text{new}}$. We then select reappearing nodes with the memory module, instantiate new nodes, and reconstruct $\widehat{\mathcal{G}}_{t+1}$ by rolling out the four decoding phases autoregressively.

Computational costs. The dominant per-snapshot computational cost of TopoGED is $O\Big(M|\widehat{\mathcal{V}}_{t+1}|s(d) + L|\widehat{\mathcal{E}}_{t+1}|d\Big)$, where M is the number of sampled candidate partners per node, $s(d)$ is the cost of scoring one edge at embedding dimension d, and L is the number of GCN layers; thus edge scoring and message passing dominate runtime. We detail the cost breakdown in App. E.4.

5 Experiments

Given a sequence of snapshots $\{\mathcal{G}_t\}_{t=1}^T$, we forecast the next snapshot $\widehat{\mathcal{G}}_{t+1}$ using information up to time t, and then evaluate downstream performance with a *forecast-then-predict* pipeline.

Datasets. We evaluate TopoGED on 14 temporal interaction benchmarks: College Message [26], MathOverflow [27] and Reddit-Body [22], TGBL-Wiki [15] and 10 ERC20 token transaction networks extracted from the Ethereum blockchain [34]. The ERC20 datasets exhibit substantial

Table 2: Summary of dataset statistics. Avg % metrics represent independent averages across all snapshots; totals may not sum to 100% due to outlier variance and temporal skew.

Dataset	Num Graphs	Avg |V|	Avg |E|	Avg % $\mathcal{V}^{\text{old}}$	Avg % oo-bank	Avg % oo-nobank	Avg % on	Avg % nn
CollegeMsg	181	123	185	91	49	39	10	3
MathOverflow	189	142	142	82	28	43	22	6
Adex	299	249	200	51	28	10	52	9
Aeternity	235	303	282	54	34	12	43	11
Aion	196	285	239	47	24	13	49	14
Aragon	343	336	309	45	18	4	74	4
Bancor	317	275	224	47	24	9	54	13
Centra	264	264	252	49	23	14	55	8
Cindicator	226	300	264	48	24	13	47	16
Coindash	274	282	267	58	31	14	52	3
DGD	725	69	61	61	38	10	47	5
Iconomi	548	179	158	55	34	8	54	5
Reddit_B	405	403	272	87	43	38	17	2
TGBL-Wiki	31	1451	1284	78	53	21	20	6

Table 3: **Forecast results.** For each dataset and metric group, we report TopoGED and the strongest baseline. Structure metric wins is the number of times each model was closest to 0 percent error across assortativity coefficient, clustering coefficient, average degree, density, triangle count, extra edges, extra nodes, missing edges, and missing nodes. Node metrics display the F1 of all nodes. Edge metrics are the F1 scores across all edge types. Detailed results can be seen in Appendix F and E.2.

Dataset	Structure metric wins ↑		Node metrics ↑		Edge metrics ↑	
	TopoGED	Best baseline	TopoGED	Best baseline	TopoGED	Best baseline
CollegeMsg	8	4 (EvolveGCN)	0.44 ± 0.10	0.04 ± 0.02 (EvolveGCN)	0.19 ± 0.07	0.00 ± 0.00 (multiple)
MathOverflow	9	2 (multiple)	0.41 ± 0.04	0.01 ± 0.01 (multiple)	0.02 ± 0.01	0.00 ± 0.00 (multiple)
Adex	6	4 (HTGN)	0.54 ± 0.09	0.09 ± 0.02 (GCLSTM)	0.09 ± 0.04	0.01 ± 0.00 (GCLSTM)
Aeternity	9	2 (multiple)	0.46 ± 0.11	0.01 ± 0.01 (ROLAND)	0.11 ± 0.04	0.00 ± 0.00 (multiple)
Aion	6	3 (multiple)	0.45 ± 0.05	0.01 ± 0.00 (GCLSTM)	0.15 ± 0.05	0.00 ± 0.00 (multiple)
Aragon	6	4 (HTGN)	0.51 ± 0.04	0.01 ± 0.00 (GCLSTM)	0.22 ± 0.08	0.00 ± 0.00 (multiple)
Bancor	6	4 (HTGN)	0.57 ± 0.07	0.03 ± 0.01 (GCLSTM)	0.14 ± 0.07	0.00 ± 0.00 (multiple)
Centra	5	4 (ROLAND)	0.47 ± 0.14	0.01 ± 0.01 (multiple)	0.07 ± 0.06	0.00 ± 0.00 (multiple)
Cindicator	6	4 (ROLAND)	0.51 ± 0.04	0.02 ± 0.01 (GCLSTM)	0.13 ± 0.07	0.00 ± 0.00 (multiple)
Coindash	5	4 (HTGN)	0.47 ± 0.12	0.03 ± 0.01 (GCLSTM)	0.13 ± 0.06	0.00 ± 0.00 (multiple)
DGD	8	3 (EvolveGCN)	0.53 ± 0.09	0.02 ± 0.01 (GCLSTM)	0.08 ± 0.05	0.00 ± 0.00 (multiple)
Iconomi	6	3 (multiple)	0.51 ± 0.09	0.01 ± 0.01 (ROLAND)	0.10 ± 0.06	0.00 ± 0.00 (multiple)
Reddit-B	8	2 (multiple)	0.38 ± 0.03	0.00 ± 0.00 (multiple)	0.03 ± 0.02	0.00 ± 0.00 (multiple)
TGBL-Wiki	7	4 (multiple)	0.65 ± 0.01	0.35 ± 0.01 (HTGN)	0.18 ± 0.01	0.01 ± 0.00 (HTGN)
Macro avg.	6.78 ± 1.32	3.35 ± 0.81 (HTGN)	0.49 ± 0.07	0.05 ± 0.09 (GCLSTM)	0.12 ± 0.06	0.00 ± 0.01 (multiple)

node churn, providing a challenging testbed for inductive forecasting. As we show in App.F, edge-type distributions shift markedly over time. Table 2 summarizes dataset statistics.

Temporal predictors. For predicting $\Phi(\widehat{\mathcal{G}}_{t+1})$ and probabilities, we evaluated SMA [47], VAR [36], V-EWMA [2], SSM [17], and VECM [16]. We adopted V-EWMA because it showed a 0.50% improvement over the nearest competitor in node-count prediction and a 0.88% improvement in edge-count prediction (See App. E.3).

Edge prediction methods. We compare TopoGED against prominent and recent temporal GNNs: ROLAND [45], EvolveGCN [28], VGRNN [11], GC-LSTM [5], and HTGN [43]. App. E.1 provides further information about these models. We train models for 25 to 100 epochs, with early stopping based on validation loss and a patience of 10 epochs. We use binary cross-entropy loss for the multi-head link prediction task. We optimize using Adam [20]. Each model undergoes hyperparameter grid search, and we select the best parameters based on validation AUC over 3 independent runs. We discuss modifications to the temporal GNNs, including parameter tuning, in Appendix C.5. We partition each dataset into 70% for training, 15% for validation, and 15% for testing.

Computational Environment. Experiments were conducted on a cluster with NVIDIA H200 NVL GPU (141 GB) nodes, dual 64-core AMD EPYC 9555 processors (128 physical cores), and 1.5 TB of system RAM.

Reproducibility. Our code is available at https://github.com/FDataLab/TopoGED.

Hyperparameters. We train all neural models with Adam, starting with a learning rate of 10^{-3} and early stopping on validation performance with patience 10. We select the edge decoder model using validation ROC-AUC. We tune all models by grid search on the same validation window. TopoGED uses history window $k = 5$, batch size 32, hidden dimension 128, one GCN layer, dropout 0.1, and weight decay 0. We evaluate each task over 3 independent runs.

Metrics. We report results in four stages. First, we evaluate descriptor prediction by forecasting the terminal node and edge counts encoded in $\Phi(\mathcal{G}_{t+1})$ and compare five temporal forecasting methods in Appendix E.3. Second, we evaluate constructed graphs at the structure level using twelve metrics, including average degree, triangle count, density, clustering coefficient, and descriptor ℓ_2 error. Third, we evaluate node-level behavior using seven metrics that measure old-node reappearance and new-node count accuracy. Finally, we evaluate edge-level behavior using eleven metrics that measure edge formation across inductive types, including precision and recall for old–old edges. App. Section E.2 defines the metrics.

Graph prediction. Table 3 reports wins and aggregate scores across structure, node, and edge evaluations. TopoGED dominates structure metrics with a macro average of 6.78 wins compared to 3.35 for the strongest baseline. This advantage is consistent across datasets, with near-complete coverage on Centra and DGD. These results show that the predicted descriptor translates into accurate global properties in the constructed graphs.

At the node level, TopoGED achieves a macro average score of 0.49, while the strongest baseline reaches 0.05. The gap is stable across datasets and is largest on Bancor and DGD. Baseline node

performance is limited by the absence of explicit node-set prediction. Standard temporal graph models score edges over a fixed or sampled node universe and rely on negative sampling for AUC evaluation. This setup measures ranking quality within a candidate set but does not constrain the size or composition of $\hat{\mathcal{V}}_{t+1}$. When these models are used for full graph construction, the active node set is induced implicitly by predicted edges and can grow without control. This exploding node-set problem leads to low precision and recall for node reappearance. In contrast, TOPOGED predicts node budgets, selects reappearing nodes through the memory module, and introduces new nodes explicitly, which yields consistent improvements.

Table 4: Ablation for CollegeMsg. For all metrics, closer to 0 is better. **Bold** indicates best, underlined is second-best.

Metric	F. Probs	F. $\Phi(\widehat{\mathcal{G}}_t)$	Oracle	TOPOGED
Avg Node Degree	3.63 ± 4.91	$\mathbf{0.17 \pm 0.35}$	0.29 ± 0.09	0.36 ± 0.19
Unique Degree Count	2.32 ± 0.82	0.90 ± 0.55	$\mathbf{0.60 \pm 0.32}$	0.64 ± 0.48
Degree Centrality	$\mathbf{0.02 \pm 0.82}$	0.09 ± 0.35	0.29 ± 0.09	0.28 ± 0.24
Assortativity Coefficient	$\mathbf{-0.87 \pm 14.05}$	2.53 ± 5.82	1.59 ± 6.98	1.13 ± 7.92
Clustering Coefficient	0.75 ± 2.82	0.34 ± 1.25	$\mathbf{0.05 \pm 0.05}$	0.07 ± 0.06
Density	$\mathbf{0.02 \pm 0.82}$	0.09 ± 0.35	0.29 ± 0.09	0.28 ± 0.24
Num Triangles	7929.25 ± 31327.34	7.32 ± 6.29	$\mathbf{1.86 \pm 2.16}$	3.11 ± 3.20
Descriptor Norm	906.68 ± 1901.83	41.74 ± 10.11	$\mathbf{27.67 \pm 7.88}$	35.42 ± 11.91
Median Extra Nodes	123.00 ± 12.85	$\mathbf{0.00 \pm 1.15}$	$\mathbf{0.00 \pm 0.00}$	$\mathbf{0.00 \pm 1.16}$
Median Missing Nodes	$\mathbf{0.00 \pm 0.00}$	0.50 ± 1.18	$\mathbf{0.00 \pm 0.00}$	0.50 ± 1.18
Median Extra Edges	413.50 ± 47.61	$\mathbf{6.00 \pm 2.80}$	8.00 ± 1.25	7.50 ± 2.67
Median Missing Edges	$\mathbf{0.00 \pm 0.00}$	$\mathbf{0.00 \pm 0.12}$	$\mathbf{0.00 \pm 0.00}$	$\mathbf{0.00 \pm 0.00}$

At the edge level, TOPOGED attains a macro average of 0.12 compared to 0.01 for the strongest baseline. The improvement appears across all datasets, with clear gains on Aragon, Coindash, and TGBL-Wiki. Baselines often achieve near-zero scores because they do not model inductive edge types and therefore fail to allocate edges across old–old, old–new, and new–new categories.

Methods such as VGRNN may appear competitive on isolated metrics by overpredicting edges, which increases recall on sampled evaluations but introduces structural noise in full-graph reconstruction. HTGN performs competitively on datasets with dominant old–old interactions such as CollegeMsg, where most edges fall into this category. This masks its limitation to old–old edges. However, it does not model node churn or inductive edge types and degrades under settings with higher turnover.

TOPOGED predicts node counts, edge counts, and all inductive edge types explicitly. It aligns node selection, edge allocation, and global structure with the predicted descriptor. This alignment yields consistent gains across structure, node, and edge evaluations and produces forecasted graphs that remain coherent under node churn and structural drift.

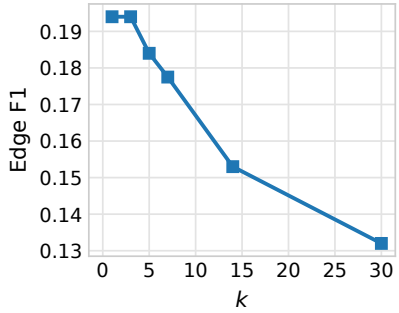


Figure 4: Sensitivity of edge prediction to history window k in TGBL-Wiki.

Ablations. We run three ablations. In *False* Φ, we replace the descriptor $\widehat{\Phi}(\mathcal{G}_{t+1})$ with a random monotone sequence $(\widetilde{X}, \widetilde{Y})$ that preserves the final node and edge counts but discards multiscale structure. In *False Probs*, we retain $\widehat{\Phi}(\mathcal{G}_{t+1})$ but replace node and edge-type probabilities with random values. In *Oracle*, we provide the decoder with the ground-truth descriptor and probabilities from $t+1$, which serves as an upper bound for reconstruction.

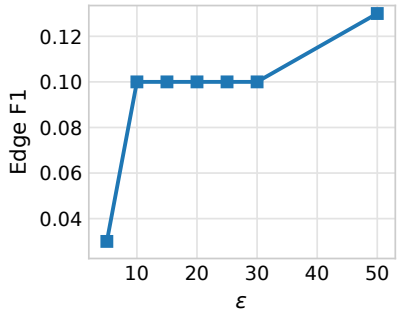


Figure 5: Sensitivity of edge prediction to the number of buckets ϵ in Adex.

Table 4 shows that each component controls a distinct failure mode. Randomizing probabilities (*F. Probs*) leads to severe over-generation, with large errors in triangle count and descriptor norm, indicating that correct edge allocation requires calibrated node and edge-type probabilities. In contrast, corrupting the descriptor (*F.* Φ) preserves budgets but distorts higher-order structure, producing large deviations in assort. and clustering coefficient. This confirms that matching final counts alone is insufficient; the multiscale profile in Φ is necessary to recover structural organization. The oracle configuration yields the lowest errors across nearly all metrics, as expected. More importantly, TOPOGED approaches this oracle without access to ground-truth future information. The gap between TOPOGED and the oracle is small across structure metrics, while both ablations

degrade substantially. This shows that accurate reconstruction requires the joint use of descriptor prediction and probabilistic decoding, and that TopoGED effectively aligns these components to recover both global structure and node-level behavior.

Efficiency. As Table 5 summarizes, TopoGED is scalable and uses substantially fewer computational resources than the baselines.

Table 5: Resource usage across 14 datasets.

Model	Time ↓ (sec)	RAM ↓ (MB)	GPU ↓ (MB)
ROLAND	319.3 ± 235.7	4512.7 ± 3315.6	1124.5 ± 569.4
TGCN	28.1 ± 19.1	6251.4 ± 3829.5	869.8 ± 528.4
GCLSTM	686.7 ± 365.2	6183.0 ± 3805.5	1534.1 ± 877.1
VGRNN	432.3 ± 269.5	3135.9 ± 1093.7	35359.2 ± 22840.8
HTGN	391.2 ± 377.8	5950.0 ± 3578.7	2923.0 ± 1646.9
EvolveGCN	92.0 ± 49.6	6411.0 ± 3853.3	1351.7 ± 878.8
TopoGED (ours)	$\mathbf{19.0 \pm 12.2}$	$\mathbf{2196.4 \pm 351.5}$	$\mathbf{839.0 \pm 1469.6}$

Sensitivity. Figures 6 and 4 show that the memory module is stable over a broad parameter range. The α–β sweep shows a large high-F1 region for moderate α and high β, indicating that persistence across recent snapshots is the strongest signal for node reappearance, while degree should receive a nonzero but not dominant weight. Very large α values reduce performance, since they overemphasize high-degree nodes and weaken the contribution of temporal recurrence.

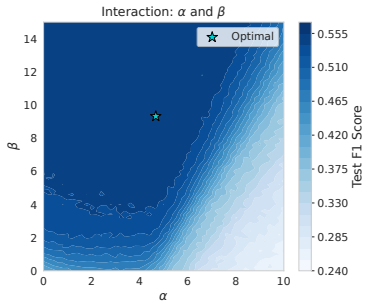


Figure 6: Sensitivity of node reappearance performance to α and β in TGBL-Wiki.

The history-window sweep shows that performance improves sharply from very short memory windows and then declines steadily as k grows. Longer windows enlarge the candidate pool which can introduce stale candidates, supporting the use of a finite recent memory for reappearing-node selection. We provide further analysis about TopoGED node predictor in Appendix C.2.

Limitations. TopoGED forecasts discrete-time snapshots rather than event-level interaction streams. Newly arriving nodes are assigned the next available ID: the model predicts the number and structural roles of new nodes without using their external identities, so it does not predict who the new nodes are before they are observed. These assumptions match the inductive snapshot-forecasting setting studied here, while continuous-time event generation and identity prediction for unseen nodes remain natural directions for future work.

6 Conclusion

We introduced TopoGED, a topology-guided encoder–decoder framework for inductive temporal graph forecasting under node churn. Rather than reducing future prediction to edge ranking over a fixed node set, TopoGED forecasts a multiscale filtration descriptor, predicts node and edge budgets, and decodes a complete future snapshot through arrival-aware edge generation.

This formulation makes node reappearance, new-node arrival, edge-type allocation, and global structural fidelity explicit parts of the prediction problem. Across diverse temporal graph benchmarks, TopoGED produces forecasted graphs that better match future node sets, edge structure, and graph-level statistics than the existing temporal graph models, while remaining computationally efficient. These results suggest that filtration-guided snapshot reconstruction is a viable alternative to purely discriminative temporal graph prediction, and provide a step toward temporal graph models that can forecast coherent future network states rather than only score isolated future interactions.

Acknowledgements

This work was partially supported by the National Science Foundation under grant DMS-2229417 and by the Simons Foundation under grant SFM-21269. The authors acknowledge the UCF Advanced Research Computing Center and Texas Advanced Computing Center (TACC) at UT Austin.

References

[1] Cuneyt Gurcan Akcora, Murat Kantarcioglu, Yulia R. Gel, and Baris Coskunuzer. Reduction algorithms for persistence diagrams of networks: Coraltda and prunit. In Sanmi Koyejo, S. Mohamed, A. Agarwal, Danielle Belgrave, K. Cho, and A. Oh, editors, *Advances*

in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022, 2022. URL http://papers.nips.cc/paper_files/paper/2022/hash/9f81a6e7081497b2d458689a4ce39fc7-Abstract-Conference.html.

[2] Robert Goodell Brown. *Statistical forecasting for inventory control*. McGraw-Hill, 1959.

[3] Mathieu Carrière, Frédéric Chazal, Yuichi Ike, Théo Lacombe, Martin Royer, and Yuhei Umeda. Perslay: A neural network layer for persistence diagrams and new graph topological signatures. In Silvia Chiappa and Roberto Calandra, editors, *The 23rd International Conference on Artificial Intelligence and Statistics, AISTATS 2020, 26-28 August 2020, Online [Palermo, Sicily, Italy]*, volume 108 of *Proceedings of Machine Learning Research*, pages 2786–2796. PMLR, 2020. URL http://proceedings.mlr.press/v108/carriere20a.html.

[4] Frédéric Chazal and Bertrand Michel. An introduction to topological data analysis: Fundamental and practical aspects for data scientists. *Frontiers Artif. Intell.*, 4:667963, 2021. doi: 10.3389/FRAI.2021.667963. URL https://doi.org/10.3389/frai.2021.667963.

[5] Jinyin Chen, Xueke Wang, and Xuanheng Xu. GC-LSTM: graph convolution embedded LSTM for dynamic network link prediction. *Appl. Intell.*, 52(7):7513–7528, 2022. doi: 10.1007/S10489-021-02518-9. URL https://doi.org/10.1007/s10489-021-02518-9.

[6] Yuzhou Chen, Baris Coskunuzer, and Yulia R. Gel. Topological relational learning on graphs. In Marc'Aurelio Ranzato, Alina Beygelzimer, Yann N. Dauphin, Percy Liang, and Jennifer Wortman Vaughan, editors, *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 27029–27042, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/e334fd9dac68f13fa1a57796148cf812-Abstract.html.

[7] Tamal Krishna Dey and Yusu Wang. *Computational topology for data analysis*. Cambridge University Press, 2022.

[8] ZhengZhao Feng, Rui Wang, TianXing Wang, Mingli Song, Sai Wu, and Shuibing He. A comprehensive survey of dynamic graph neural networks: Models, frameworks, benchmarks, experiments and challenges. *IEEE Trans. Knowl. Data Eng.*, 38(1):26–46, 2026. doi: 10.1109/TKDE.2025.3621291. URL https://doi.org/10.1109/TKDE.2025.3621291.

[9] Alessio Gravina, Giulio Lovisotto, Claudio Gallicchio, Davide Bacciu, and Claas Grohnfeldt. Long range propagation on continuous-time dynamic graphs. In Ruslan Salakhutdinov, Zico Kolter, Katherine A. Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp, editors, *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*, volume 235 of *Proceedings of Machine Learning Research*, pages 16206–16225. PMLR / OpenReview.net, 2024. URL https://proceedings.mlr.press/v235/gravina24a.html.

[10] Shubham Gupta, Sahil Manchanda, Srikanta Bedathur, and Sayan Ranu. TIGGER: scalable generative modelling for temporal interaction graphs. In *Thirty-Sixth AAAI Conference on Artificial Intelligence, AAAI 2022, Thirty-Fourth Conference on Innovative Applications of Artificial Intelligence, IAAI 2022, The Twelveth Symposium on Educational Advances in Artificial Intelligence, EAAI 2022 Virtual Event, February 22 - March 1, 2022*, pages 6819–6828. AAAI Press, 2022. doi: 10.1609/AAAI.V36I6.20638. URL https://doi.org/10.1609/aaai.v36i6.20638.

[11] Ehsan Hajiramezanali, Arman Hasanzadeh, Krishna R. Narayanan, Nick Duffield, Mingyuan Zhou, and Xiaoning Qian. Variational graph recurrent neural networks. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 10700–10710, 2019. URL https://proceedings.neurips.cc/paper/2019/hash/a6b8deb7798e7532ade2a8934477d3ce-Abstract.html.

[12] Christoph D. Hofer, Florian Graf, Bastian Rieck, Marc Niethammer, and Roland Kwitt. Graph filtration learning. In *Proceedings of the 37th International Conference on Machine Learning,*

ICML 2020, 13-18 July 2020, Virtual Event, volume 119 of *Proceedings of Machine Learning Research*, pages 4314–4323. PMLR, 2020. URL http://proceedings.mlr.press/v119/hofer20b.html.

[13] Max Horn, Edward De Brouwer, Michael Moor, Yves Moreau, Bastian Rieck, and Karsten M. Borgwardt. Topological graph neural networks. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022. URL https://openreview.net/forum?id=oxxUMeFwEHd.

[14] Ryien Hosseini, Filippo Simini, Venkatram Vishwanath, and Henry Hoffmann. A deep probabilistic framework for continuous time dynamic graph generation. In Toby Walsh, Julie Shah, and Zico Kolter, editors, *Thirty-Ninth AAAI Conference on Artificial Intelligence, Thirty-Seventh Conference on Innovative Applications of Artificial Intelligence, Fifteenth Symposium on Educational Advances in Artificial Intelligence, AAAI 2025, Philadelphia, PA, USA, February 25 - March 4, 2025*, pages 17249–17257. AAAI Press, 2025. doi: 10.1609/AAAI.V39I16.33896. URL https://doi.org/10.1609/aaai.v39i16.33896.

[15] Shenyang Huang, Farimah Poursafaei, Jacob Danovitch, Matthias Fey, Weihua Hu, Emanuele Rossi, Jure Leskovec, Michael M. Bronstein, Guillaume Rabusseau, and Reihaneh Rabbany. Temporal graph benchmark for machine learning on temporal graphs. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine, editors, *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023. URL http://papers.nips.cc/paper_files/paper/2023/hash/066b98e63313162f6562b35962671288-Abstract-Datasets_and_Benchmarks.html.

[16] Søren Johansen. Estimation and hypothesis testing of cointegration vectors in gaussian vector autoregressive models. *Econometrica: Journal of the Econometric Society*, pages 1551–1580, 1991.

[17] Rudolf Emil Kalman. A new approach to linear filtering and prediction problems. *Journal of Basic Engineering*, 82(1):35–45, 1960.

[18] Yannis Karmim, Marc Lafon, Raphaël Fournier-S'niehotta, and Nicolas Thome. Supra-laplacian encoding for transformer on dynamic graphs. In Amir Globersons, Lester Mackey, Danielle Belgrave, Angela Fan, Ulrich Paquet, Jakub M. Tomczak, and Cheng Zhang, editors, *Advances in Neural Information Processing Systems 37: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, 2024. URL http://papers.nips.cc/paper_files/paper/2024/hash/1ed4723f12853cbd02aecb8160f5e0c9-Abstract-Conference.html.

[19] Seyed Mehran Kazemi. Dynamic graph neural networks. In *Graph neural networks: Foundations, frontiers, and applications*, pages 323–349. Springer, 2022.

[20] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. doi: 10.48550/ARXIV.1412.6980. URL https://doi.org/10.48550/arXiv.1412.6980.

[21] Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017. URL https://openreview.net/forum?id=SJU4ayYgl.

[22] Srijan Kumar, William L. Hamilton, Jure Leskovec, and Dan Jurafsky. Community interaction and conflict on the web. In Pierre-Antoine Champin, Fabien Gandon, Mounia Lalmas, and Panagiotis G. Ipeirotis, editors, *Proceedings of the 2018 World Wide Web Conference on World Wide Web, WWW 2018, Lyon, France, April 23-27, 2018*, pages 933–943. ACM, 2018. doi: 10.1145/3178876.3186141. URL https://doi.org/10.1145/3178876.3186141.

[23] Srijan Kumar, Xikun Zhang, and Jure Leskovec. Predicting dynamic embedding trajectory in temporal interaction networks. In Ankur Teredesai, Vipin Kumar, Ying Li, Rómer Rosales,

Evimaria Terzi, and George Karypis, editors, *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2019, Anchorage, AK, USA, August 4-8, 2019*, pages 1269–1278. ACM, 2019. doi: 10.1145/3292500.3330895. URL https://doi.org/10.1145/3292500.3330895.

[24] Fan Li, Xiaoyang Wang, Dawei Cheng, Cong Chen, Ying Zhang, and Xuemin Lin. Efficient dynamic attributed graph generation. In *41st IEEE International Conference on Data Engineering, ICDE 2025, Hong Kong, May 19-23, 2025*, pages 1415–1428. IEEE, 2025. doi: 10.1109/ICDE65448.2025.00110. URL https://doi.org/10.1109/ICDE65448.2025.00110.

[25] Xiao Luo, Jingyang Yuan, Zijie Huang, Huiyu Jiang, Yifang Qin, Wei Ju, Ming Zhang, and Yizhou Sun. HOPE: high-order graph ODE for modeling interacting dynamics. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett, editors, *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pages 23124–23139. PMLR, 2023. URL https://proceedings.mlr.press/v202/luo23f.html.

[26] Pietro Panzarasa, Tore Opsahl, and Kathleen M. Carley. Patterns and dynamics of users' behavior and interaction: Network analysis of an online community. *J. Assoc. Inf. Sci. Technol.*, 60(5): 911–932, 2009. doi: 10.1002/ASI.21015. URL https://doi.org/10.1002/asi.21015.

[27] Ashwin Paranjape, Austin R. Benson, and Jure Leskovec. Motifs in temporal networks. In Maarten de Rijke, Milad Shokouhi, Andrew Tomkins, and Min Zhang, editors, *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining, WSDM 2017, Cambridge, United Kingdom, February 6-10, 2017*, pages 601–610. ACM, 2017. doi: 10.1145/3018661.3018731. URL https://doi.org/10.1145/3018661.3018731.

[28] Aldo Pareja, Giacomo Domeniconi, Jie Chen, Tengfei Ma, Toyotaro Suzumura, Hiroki Kanezashi, Tim Kaler, Tao B. Schardl, and Charles E. Leiserson. Evolvegcn: Evolving graph convolutional networks for dynamic graphs. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 5363–5370. AAAI Press, 2020. doi: 10.1609/AAAI.V34I04.5984. URL https://doi.org/10.1609/aaai.v34i04.5984.

[29] Joonhyuk Park, Donghyun Lee, Yujee Song, Guorong Wu, and Won Hwa Kim. Topology-aware graph diffusion model with persistent homology. In Danielle Belgrave, Cheng Zhang, Laura N. Montoya, Hsuan-Tien Lin, Razvan Pascanu, Piotr Koniusz, Marzyeh Ghassemi, Nancy Chen, Iván Vladimir Meza Ruíz, and Arturo Loaiza-Bonilla, editors, *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2025, NeurIPS 2025, San Diego, CA, USA, December 2-7, 2025 / Mexico City, Mexico, November 30 - December 5, 2025*, 2025. URL http://papers.nips.cc/paper_files/paper/2025/hash/bb88dfbebcb21022d32086bed631bfc5-Abstract-Conference.html.

[30] Farimah Poursafaei, Shenyang Huang, Kellin Pelrine, and Reihaneh Rabbany. Towards better evaluation for dynamic link prediction. In Sanmi Koyejo, S. Mohamed, A. Agarwal, Danielle Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, 2022. URL http://papers.nips.cc/paper_files/paper/2022/hash/d49042a5d49818711c401d34172f9900-Abstract-Datasets_and_Benchmarks.html.

[31] Emanuele Rossi, Ben Chamberlain, Fabrizio Frasca, Davide Eynard, Federico Monti, and Michael M. Bronstein. Temporal graph networks for deep learning on dynamic graphs. *CoRR*, abs/2006.10637, 2020. doi: 10.48550/ARXIV.2006.10637. URL https://doi.org/10.48550/arXiv.2006.10637.

[32] Aravind Sankar, Yanhong Wu, Liang Gou, Wei Zhang, and Hao Yang. Dysat: Deep neural representation learning on dynamic graphs via self-attention networks. In James Caverlee, Xia (Ben) Hu, Mounia Lalmas, and Wei Wang, editors, *WSDM '20: The Thirteenth ACM International Conference on Web Search and Data Mining, Houston, TX, USA, February 3-7,*

2020, pages 519–527. ACM, 2020. doi: 10.1145/3336191.3371845. URL https://doi.org/10.1145/3336191.3371845.

[33] Kiarash Shamsi, Friedhelm Victor, Murat Kantarcioglu, Yulia R. Gel, and Cuneyt Gurcan Akcora. Chartalist: Labeled graph datasets for UTXO and account-based blockchains. In Sanmi Koyejo, S. Mohamed, A. Agarwal, Danielle Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, 2022. URL http://papers.nips.cc/paper_files/paper/2022/hash/e245189a86310b6667ac633dbb922d50-Abstract-Datasets_and_Benchmarks.html.

[34] Kiarash Shamsi, Farimah Poursafaei, Shenyang Huang, Tran Gia Bao Ngo, Baris Coskunuzer, and Cuneyt Gurcan Akcora. Graphpulse: Topological representations for temporal graph property prediction. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=DZqic2sPTY.

[35] Kiarash Shamsi, Tran Gia Bao Ngo, Razieh Shirzadkhani, Shenyang Huang, Farimah Poursafaei, Poupak Azad, Reihaneh Rabbany, Baris Coskunuzer, Guillaume Rabusseau, and Cuneyt Gurcan Akcora. Mint: Multi-network transfer benchmark for temporal graph learning. In Danielle Belgrave, Cheng Zhang, Laura N. Montoya, Hsuan-Tien Lin, Razvan Pascanu, Piotr Koniusz, Marzyeh Ghassemi, Nancy Chen, Iván Vladimir Meza Ruíz, and Arturo Loaiza-Bonilla, editors, *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2025, NeurIPS 2025, San Diego, CA, USA, December 2-7, 2025 / Mexico City, Mexico, November 30 - December 5, 2025*, 2025. URL http://papers.nips.cc/paper_files/paper/2025/hash/c5548168cb7324f714365a971dfe76d1-Abstract-Datasets_and_Benchmarks_Track.html.

[36] Christopher A. Sims. Macroeconomics and reality. *Econometrica: Journal of the Econometric Society*, pages 1–48, 1980.

[37] Junwei Su, Difan Zou, and Chuan Wu. PRES: toward scalable memory-based dynamic graph neural networks. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=gjXor87Xfy.

[38] Yuxing Tian, Yiyan Qi, and Fan Guo. Freedyg: Frequency enhanced continuous-time dynamic graph model for link prediction. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=82Mc5ilInM.

[39] Astrit Tola, Funmilola Mary Taiwo, Cuneyt Gurcan Akcora, and Baris Coskunuzer. Toper: Topological embeddings in graph representation learning. In Danielle Belgrave, Cheng Zhang, Laura N. Montoya, Hsuan-Tien Lin, Razvan Pascanu, Piotr Koniusz, Marzyeh Ghassemi, Nancy Chen, Iván Vladimir Meza Ruíz, and Arturo Loaiza-Bonilla, editors, *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2025, NeurIPS 2025, San Diego, CA, USA, December 2-7, 2025 / Mexico City, Mexico, November 30 - December 5, 2025*, 2025. URL http://papers.nips.cc/paper_files/paper/2025/hash/8027ace5713843619206665f1d1b69758-Abstract-Conference.html.

[40] Rakshit S. Trivedi, Mehrdad Farajtabar, Prasenjeet Biswal, and Hongyuan Zha. Dyrep: Learning representations over dynamic graphs. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019. URL https://openreview.net/forum?id=HyePrhR5KX.

[41] Ronald J. Williams and David Zipser. A learning algorithm for continually running fully recurrent neural networks. *Neural Comput.*, 1(2):270–280, 1989. doi: 10.1162/NECO.1989.1.2.270. URL https://doi.org/10.1162/neco.1989.1.2.270.

[42] Da Xu, Chuanwei Ruan, Evren Körpeoglu, Sushant Kumar, and Kannan Achan. Inductive representation learning on temporal graphs. In *8th International Conference on Learning*

Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020. OpenReview.net, 2020. URL https://openreview.net/forum?id=rJeW1yHYwH.

[43] Menglin Yang, Min Zhou, Marcus Kalander, Zengfeng Huang, and Irwin King. Discrete-time temporal network embedding via implicit hierarchical learning in hyperbolic space. In Feida Zhu, Beng Chin Ooi, and Chunyan Miao, editors, *KDD '21: The 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Virtual Event, Singapore, August 14-18, 2021*, pages 1975–1985. ACM, 2021. doi: 10.1145/3447548.3467422. URL https://doi.org/10.1145/3447548.3467422.

[44] Dongsheng Ye, Hao Jiang, Ying Jiang, and Hao Li. Stable distance of persistent homology for dynamic graph comparison. *Knowl. Based Syst.*, 278:110855, 2023. doi: 10.1016/J.KNOSYS.2023.110855. URL https://doi.org/10.1016/j.knosys.2023.110855.

[45] Jiaxuan You, Tianyu Du, and Jure Leskovec. ROLAND: graph learning framework for dynamic graphs. In Aidong Zhang and Huzefa Rangwala, editors, *KDD '22: The 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Washington, DC, USA, August 14 - 18, 2022*, pages 2358–2366. ACM, 2022. doi: 10.1145/3534678.3539300. URL https://doi.org/10.1145/3534678.3539300.

[46] Xingtong Yu, Zhenghao Liu, Xinming Zhang, and Yuan Fang. Node-time conditional prompt learning in dynamic graphs. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025. URL https://openreview.net/forum?id=kVlfYvIqaK.

[47] G. Udny Yule. On the time-correlation problem, with especial reference to the variate-difference correlation method. *Journal of the Royal Statistical Society*, 84(4):497–537, 1921.

[48] Wenbin Zhang, Liming Zhang, Dieter Pfoser, and Liang Zhao. Disentangled dynamic graph deep generation. In Carlotta Demeniconi and Ian Davidson, editors, *Proceedings of the 2021 SIAM International Conference on Data Mining, SDM 2021, Virtual Event, April 29 - May 1, 2021*, pages 738–746. SIAM, 2021. doi: 10.1137/1.9781611976700.83. URL https://doi.org/10.1137/1.9781611976700.83.

[49] Ling Zhao, Yujiao Song, Chao Zhang, Yu Liu, Pu Wang, Tao Lin, Min Deng, and Haifeng Li. T-GCN: A temporal graph convolutional network for traffic prediction. *IEEE Trans. Intell. Transp. Syst.*, 21(9):3848–3858, 2020. doi: 10.1109/TITS.2019.2935152. URL https://doi.org/10.1109/TITS.2019.2935152.

[50] Yanping Zheng, Lu Yi, and Zhewei Wei. A survey of dynamic graph neural networks. *Frontiers Comput. Sci.*, 19(6):196323, 2025. doi: 10.1007/S11704-024-3853-2. URL https://doi.org/10.1007/s11704-024-3853-2.

[51] Dawei Zhou, Lecheng Zheng, Jiawei Han, and Jingrui He. A data-driven graph generative model for temporal interaction networks. In Rajesh Gupta, Yan Liu, Jiliang Tang, and B. Aditya Prakash, editors, *KDD '20: The 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Virtual Event, CA, USA, August 23-27, 2020*, pages 401–411. ACM, 2020. doi: 10.1145/3394486.3403082. URL https://doi.org/10.1145/3394486.3403082.

[52] Jason Zhu, Arijit Khan, and Cuneyt Gurcan Akcora. Data depth and core-based trend detection on blockchain transaction networks. *Frontiers Blockchain*, 7, 2024. doi: 10.3389/FBLOC.2024.1342956. URL https://doi.org/10.3389/fbloc.2024.1342956.

Appendix

A Broader Impact

TopoGED can broaden the use of temporal graph learning in settings where decisions must be made before the next network state is observed. By forecasting an entire inductive snapshot, including node arrivals, reappearances, and future edge structure, it supports early risk assessment, anomaly monitoring, resource planning, and downstream analysis in evolving networks such as transaction ecosystems, online communities, and scientific interaction graphs. Its topology-guided design makes forecasts more interpretable than isolated edge scores, since users can inspect predicted node counts, edge budgets, churn, and structural properties before applying downstream models. Because the output is a full forecasted graph, multiple tasks can be evaluated on the same predicted snapshot without task-specific retraining. The efficient descriptor-based pipeline also reduces computational cost, making graph forecasting more accessible for large dynamic networks.

B Extended Related Work

Temporal graph representation learning. Temporal graph models learn time-evolving node and edge representations to predict links, nodes, or graph-level labels. Early methods update embeddings along interaction streams with memory, e.g., JODIE [23] and DyRep [40]. More recent temporal GNNs incorporate temporal encoding and attention to support inductive learning over time-stamped edges, e.g., TGAT [42] and TGN [31], while snapshot-based models attend over sequences of graphs, e.g., DySAT [32]. PRES is an example of a dynamic memory model that optimizes for efficient training between snapshots [37].

CTDG Methods. CTDGs [38, 25, 45] model the network as a stream of time-stamped node-level events. Memory-based CTDG methods update embeddings along interaction streams [23, 40], while temporal GNNs add attention and temporal encoding for inductive learning [42, 31].

DTDG Methods.DTDGs represent the network as a sequence of snapshot graphs collected at discrete timestamps. Snapshot-based models attend over sequences of graphs, e.g., DySAT [32]. PRES is an example of a dynamic memory model that optimizes for efficient training between snapshots [37]. HTGN [43] project each snapshot into hyperbolic space to model node representation. While existing methods focus on node-level tasks, GraphPulse [34] and MiNT [35] are snapshot-based models for graph property prediction tasks.

Dynamic graph generation. Dynamic graph generation has been studied in both discrete-time and continuous-time regimes. In discrete time, prior methods generate sequences of snapshots by learning recurrent or variational snapshot latents [48, 24]. In continuous time, event-based generators model timestamped interactions, often via temporal walks or point-process machinery [9, 51, 10, 14]; while effective for edge-event synthesis, they do not directly yield coherent snapshot forecasts under churn and are commonly tied to training node IDs. This leaves a gap for discrete-time snapshot forecasting methods that can inductively predict future graphs with new nodes while maintaining global structural fidelity.

C Forecast-to-Graph Reconstruction Details

This section provides the implementation details for converting the quantities forecast by TopoGED into a concrete future snapshot $\widehat{\mathcal{G}}_{t+1} = (\widehat{\mathcal{V}}_{t+1}, \widehat{\mathcal{E}}_{t+1})$. We first describe the degree-based filtration descriptor and its inverse reconstruction role in Appendix C.1. We then detail how the predicted node set is assembled, including reappearing-node sampling from a recent memory window in Appendix C.2 and degree-bin assignment for cold-start nodes in Appendix C.3. Next, we define the edge-type probabilities used to allocate the predicted edge budget across old–old-bank, old–old-nobank, old–new, and new–new interactions in Appendix C.4. These components are combined in the full budgeted multi-phase reconstruction algorithm in Algorithm 1. Finally, Appendix C.5 explains how link-prediction baselines are adapted to the same full-graph construction setting through thresholding and bounded adjacency reconstruction.

C.1 Degree-Based Filtration Encoder and Inverse Topology Decoder

Let $\mathcal{G}_t = (\mathcal{V}_t, \mathcal{E}_t)$ denote a graph snapshot at time t, where $\mathcal{V}_t$ is the node set and $\mathcal{E}_t$ is the edge set. We encode each snapshot by a filtration-based multiscale topological descriptor. The goal of this descriptor is to summarize the evolution of the graph structure across increasing degree thresholds, while retaining enough information to later reconstruct a graph snapshot consistent with the encoded topology.

We use the degree function

$$f_t(v) = \deg_{\mathcal{G}_t}(v), \qquad v \in \mathcal{V}_t,$$

as the filtration function. Given degree thresholds

$$\epsilon_1 < \epsilon_2 < \cdots < \epsilon_n,$$

we define the active vertex set at filtration level ϵ_i by

$$\mathcal{V}_{t,\epsilon_i} = \{v \in \mathcal{V}_t : f_t(v) \le \epsilon_i\}.$$

The corresponding filtered graph is the induced subgraph

$$\mathcal{G}_{t,\epsilon_i} = \mathcal{G}_t[\mathcal{V}_{t,\epsilon_i}] = (\mathcal{V}_{t,\epsilon_i}, \mathcal{E}_{t,\epsilon_i}),$$

where

$$\mathcal{E}_{t,\epsilon_i} = \{(u,v) \in \mathcal{E}_t : u, v \in \mathcal{V}_{t,\epsilon_i}\}.$$

Thus, the filtration produces a nested sequence of induced subgraphs

$$\mathcal{G}_{t,\epsilon_1} \subseteq \mathcal{G}_{t,\epsilon_2} \subseteq \cdots \subseteq \mathcal{G}_{t,\epsilon_n} = \mathcal{G}_t.$$

At each filtration level, we record two structural quantities: the number of active vertices and the number of induced edges. Specifically, we define

$$x_{t,i} = |\mathcal{V}_{t,\epsilon_i}|, \qquad y_{t,i} = |\mathcal{E}_{t,\epsilon_i}|.$$

This gives two monotone sequences

$$X_t = (x_{t,1}, x_{t,2}, \ldots, x_{t,n})$$

and

$$Y_t = (y_{t,1}, y_{t,2}, \ldots, y_{t,n}).$$

The filtration-based topological descriptor of the snapshot $\mathcal{G}_t$ is then defined as

$$\Phi(\mathcal{G}_t) \coloneqq (X_t, Y_t).$$

By construction, the final filtration level recovers the size of the original snapshot:

$$x_{t,n} = |\mathcal{V}_t|, \qquad y_{t,n} = |\mathcal{E}_t|.$$

Therefore, $\Phi(\mathcal{G}_t)$ provides a compact multiscale representation of the snapshot, recording how the number of vertices and induced edges grows as the degree threshold increases.

The use of the degree filtration has two advantages. First, degree is computationally inexpensive and directly available from the graph structure. Second, and more importantly, degree provides a natural handle on the inverse topology problem. In the forward direction, the encoder maps a graph snapshot to its descriptor,

$$\mathcal{G}_t \longmapsto \Phi(\mathcal{G}_t) = (X_t, Y_t).$$

The inverse topology problem asks for the reverse operation: given a predicted descriptor

$$\widehat{\Phi}_{t+1} = (\widehat{X}_{t+1}, \widehat{Y}_{t+1}),$$

construct a graph snapshot

$$\widehat{\mathcal{G}}_{t+1} = (\widehat{\mathcal{V}}_{t+1}, \widehat{\mathcal{E}}_{t+1})$$

whose filtration realizes the predicted descriptor. That is, we seek

$$\Phi(\widehat{\mathcal{G}}_{t+1}) \approx \widehat{\Phi}_{t+1}.$$

Equivalently, for each threshold ϵ_i, the induced subgraph of $\widehat{\mathcal{G}}_{t+1}$ should satisfy

$$|\widehat{\mathcal{V}}_{t+1,\epsilon_i}| = \widehat{x}_{t+1,i}, \qquad |\widehat{\mathcal{E}}_{t+1,\epsilon_i}| = \widehat{y}_{t+1,i}.$$

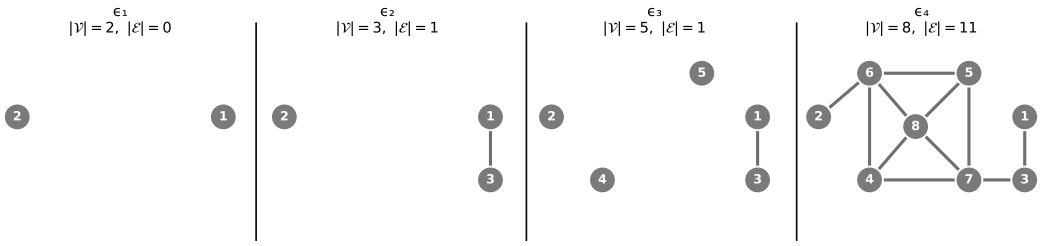


Figure 7: Degree-threshold filtration of a graph snapshot. At each filtration level ϵ_i, the induced subgraph records the number of active vertices $|\mathcal{V}|$ and induced edges $|\mathcal{E}|$.

This inverse problem should be understood as a graph realization problem rather than exact graph recovery. The descriptor (X_t, Y_t) does not encode the full adjacency matrix of $\mathcal{G}_t$; instead, it records the number of active vertices and induced edges at each degree threshold. Consequently, many non-isomorphic graphs may share the same descriptor. The decoder therefore aims to construct one graph that is consistent with the predicted multiscale topology.

A natural decoding procedure proceeds level by level through the predicted filtration. Suppose the predicted descriptor is

$$\widehat{X} = (\widehat{x}_1, \dots, \widehat{x}_n), \qquad \widehat{Y} = (\widehat{y}_1, \dots, \widehat{y}_n).$$

At the first filtration level, the decoder creates $\widehat{x}_1$ vertices and adds $\widehat{y}_1$ induced edges among them. At the next level, it adds

$$\widehat{x}_2 - \widehat{x}_1$$

new vertices and then adds edges so that the induced edge count becomes $\widehat{y}_2$. Continuing in this way, at level i the decoder adds

$$\widehat{x}_i - \widehat{x}_{i-1}$$

new vertices and introduces

$$\widehat{y}_i - \widehat{y}_{i-1}$$

additional induced edges, subject to the constraint that all previously constructed subgraphs remain unchanged. Thus, the predicted descriptor controls both node inclusion and edge growth across the filtration.

Formally, the decoder constructs nested vertex sets

$$\widehat{\mathcal{V}}_{\epsilon_1} \subseteq \widehat{\mathcal{V}}_{\epsilon_2} \subseteq \cdots \subseteq \widehat{\mathcal{V}}_{\epsilon_n} = \widehat{\mathcal{V}}$$

with

$$|\widehat{\mathcal{V}}_{\epsilon_i}| = \widehat{x}_i,$$

and edge sets satisfying

$$|\widehat{\mathcal{E}}_{\epsilon_i}| = \widehat{y}_i, \qquad \widehat{\mathcal{E}}_{\epsilon_i} = \{(u, v) \in \widehat{\mathcal{E}} : u, v \in \widehat{\mathcal{V}}_{\epsilon_i}\}.$$

The final output is the graph

$$\widehat{\mathcal{G}} = (\widehat{\mathcal{V}}, \widehat{\mathcal{E}}),$$

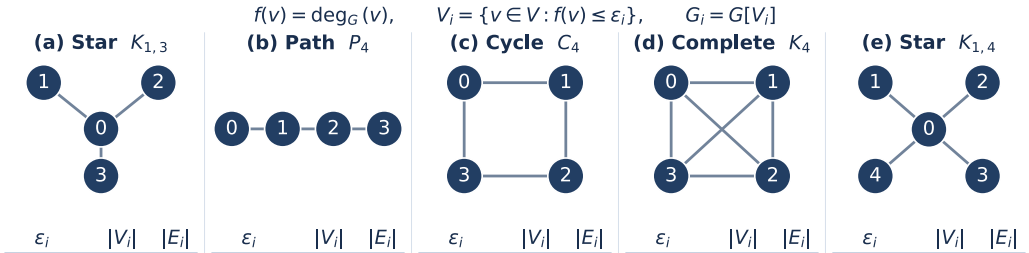
where

$$|\widehat{\mathcal{V}}| = \widehat{x}_n, \qquad |\widehat{\mathcal{E}}| = \widehat{y}_n.$$

This construction makes the descriptor directly actionable. The sequence $\widehat{X}$ determines how many vertices must be introduced at each degree threshold, while the sequence $\widehat{Y}$ determines how many induced edges must be present among those vertices. Therefore, the degree-based descriptor is not only a representation of the graph; it also serves as a blueprint for constructing a graph snapshot from predicted topological data.

Figure 7 illustrates this process. For a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, the filtration levels $\epsilon_1, \epsilon_2, \epsilon_3, \epsilon_4$ produce induced subgraphs with descriptors

$$(|\mathcal{V}_{\epsilon_1}|, |\mathcal{E}_{\epsilon_1}|), \quad (|\mathcal{V}_{\epsilon_2}|, |\mathcal{E}_{\epsilon_2}|), \quad (|\mathcal{V}_{\epsilon_3}|, |\mathcal{E}_{\epsilon_3}|), \quad (|\mathcal{V}_{\epsilon_4}|, |\mathcal{E}_{\epsilon_4}|).$$



ε_i	$\lvert V_i\rvert$	$\lvert E_i\rvert$
$\varepsilon_1 = 1$	3	0
$\varepsilon_2 = 2$	3	0
$\varepsilon_3 = 3$	4	3
$\varepsilon_4 = 4$	4	3

ε_i	$\lvert V_i\rvert$	$\lvert E_i\rvert$
$\varepsilon_1 = 1$	2	0
$\varepsilon_2 = 2$	4	3
$\varepsilon_3 = 3$	4	3
$\varepsilon_4 = 4$	4	3

ε_i	$\lvert V_i\rvert$	$\lvert E_i\rvert$
$\varepsilon_1 = 1$	0	0
$\varepsilon_2 = 2$	4	4
$\varepsilon_3 = 3$	4	4
$\varepsilon_4 = 4$	4	4

ε_i	$\lvert V_i\rvert$	$\lvert E_i\rvert$
$\varepsilon_1 = 1$	0	0
$\varepsilon_2 = 2$	0	0
$\varepsilon_3 = 3$	4	6
$\varepsilon_4 = 4$	4	6

ε_i	$\lvert V_i\rvert$	$\lvert E_i\rvert$
$\varepsilon_1 = 1$	4	0
$\varepsilon_2 = 2$	4	0
$\varepsilon_3 = 3$	4	0
$\varepsilon_4 = 4$	5	4

Figure 8: Degree-based structural descriptors for the star graphs $K_{1,3}$ and $K_{1,4}$, path P_4, cycle C_4, and complete graph K_4. Each table reports the vertex count $|V_i|$ and edge count $|E_i|$ of the induced subgraph $G_i = G[\{v \in V : \deg_G(v) \le \epsilon_i\}]$ at thresholds $\epsilon_i = i$ for $i \in \{1, 2, 3, 4\}$. Node labels denote vertex identifiers, and d lists degrees in the original graph in vertex order.

In the displayed example, these values are

$$(2, 0), \qquad (3, 1), \qquad (5, 1), \qquad (8, 11).$$

Thus,

$$X = (2, 3, 5, 8), \qquad Y = (0, 1, 1, 11).$$

Reading the filtration forward encodes the graph into the descriptor $\Phi(\mathcal{G}) = (X, Y)$, while reading it in reverse shows how the same descriptor can be used to decode a graph realization. The large increase from $|\mathcal{E}_{\epsilon_3}| = 1$ to $|\mathcal{E}_{\epsilon_4}| = 11$ indicates that most of the edge structure appears when the final group of vertices enters the filtration. Hence, the filtration descriptor captures not only the final size of the graph, but also the scale at which its connectivity emerges.

In summary, the encoder maps each snapshot $\mathcal{G}_t$ to a compact degree-based topological descriptor $\Phi(\mathcal{G}_t) = (X_t, Y_t)$. The inverse topology decoder then uses a predicted descriptor to construct a graph snapshot whose thresholded induced subgraphs match the predicted node and edge counts. This provides a direct bridge between topological prediction and graph generation: instead of predicting every edge independently, the model predicts a multiscale structural summary, and the decoder realizes this summary as a concrete graph.

C.2 Node Set Prediction and Reappearance Sampling

A primary contribution of TopoGED is the node prediction heuristic for old nodes. For each timestamp T, instead of predicting edges with the entire node universe for the dataset, we assign nodes a 'reappearance probability' for weighted random sampling.

For each $v \in \mathcal{V}_{t-k:t}$, we assign

$$P(v, t) = \exp\left(-\frac{t - \ell_v}{\lambda}\right) \left(\frac{\deg_v}{\deg_{\max}}\right)^{\alpha} f_v^{\beta}, \tag{2}$$

where ℓ_v is the last observed time of v, $\deg_v$ is its degree aggregated over the window, $\deg_{\max}$ is the maximum degree in the window, and f_v is the number of snapshots in which v appears within the window. The exponential term enforces recency, the normalized degree captures structural prominence, and the frequency term captures persistence across snapshots. We keep these factors separate to avoid conflating instantaneous connectivity with temporal activity. The normalization by $\deg_{\max}$ keeps scores on a comparable scale across datasets.

According to the descriptor and probability vector, n nodes are randomly sampled and added to the current graph to be constructed.

Weighted sampling prevents the decoder from always selecting the same highest-score nodes, while still favoring nodes with stronger recency, degree, and frequency evidence.

Tables 6 and 7 summarize the fitted parameters and the resulting node reappearance performance. The optimal β values are consistently high across datasets, showing that repeated activity within

the memory window is the strongest signal for old-node reappearance. The α values vary more substantially, which indicates that degree is useful but dataset-dependent. The fitted λ values are generally below 1, so recent activity receives much larger weight than older activity in most datasets.

The F1 scores show that node reappearance is highly dataset-dependent. TGBL-Wiki and CollegeMsg achieve the strongest test performance, while sparse token graphs such as Aion and Aragon remain difficult. In several datasets, validation and test performance exceed training performance, suggesting that the heuristic is not simply memorizing the training snapshots. Overall, these results support the use of a small memory module: recency, degree, and frequency provide enough signal to select plausible old nodes without scoring the full historical node universe.

Table 6: Optimal Parameters

Dataset	α	β	λ
CollegeMsg	3.88	8.81	0.31
Adex	2.64	8.94	0.85
Aeternity	3.53	9.24	0.53
Aion	1.94	8.75	0.60
Aragon	3.53	9.99	0.55
Bancor	4.19	9.51	0.64
Centra	3.25	9.78	0.35
Cindicator	3.74	9.24	0.51
Coindash	2.97	9.47	0.63
Dgd	1.32	9.96	0.31
Iconomi	2.94	9.70	0.37
mathoverflow	5.94	7.46	0.15
Reddit_B	3.92	9.96	0.96
TGBL-Wiki	4.67	9.32	0.21

Table 7: Node reappearance prediction F1 Score

Dataset	Train	Val	Test
CollegeMsg	0.4563	0.4366	0.3976
Adex	0.0915	0.1637	0.2257
Aeternity	0.1276	0.2103	0.2353
Aion	0.0645	0.0917	0.0944
Aragon	0.0719	0.0670	0.1128
Bancor	0.1402	0.1710	0.2383
Centra	0.1230	0.2380	0.1723
Cindicator	0.0655	0.1197	0.1491
Coindash	0.1311	0.2483	0.2866
Dgd	0.2203	0.1999	0.2428
Iconomi	0.1306	0.1136	0.2212
mathoverflow	0.3160	0.3125	0.2993
Reddit_B	0.2511	0.2556	0.2777
TGBL-Wiki	0.5542	0.5608	0.5525

C.3 Node Binning and Cold-Start Initialization

For nodes $v \in \mathcal{V}_{t+1}$ in the node set $\mathcal{V}$ of graph G_{t+1}, we assign a maximum degree according to the current filtration sequence.

Previously existing nodes, $v \in \mathcal{V}_{\text{old}}$ are assigned a maximum degree equivalent to their most recent appearance, rounded up to the threshold ϵ_k for the nearest non-empty bin.

$$\deg(v) = \min\{\epsilon_k \in \mathcal{B} \mid \epsilon_k \geq \deg(v)_\tau \text{ and } V_k > 0\}$$

Where ϵ_k is the threshold for each bin in $\mathcal{B}$, $\deg(v)_\tau$ is the most recently seen degree for node v, and $\mathcal{V}_k$ is the number of nodes in bin k.

For $v \in \mathcal{V}_{\text{new}}$, new nodes initially have no information. Therefore, for each node, a maximum degree ϵ is randomly assigned if the bin is not empty.

$$\deg(v) = \text{random}(\{\epsilon_k \in \mathcal{B} \mid V_k > 0\})$$

C.4 Edge-Type Counts and Probabilities

We study the relative prevalence of inductive edge types in snapshot $\mathcal{G}_t = (\mathcal{V}_t, \mathcal{E}_t)$ using the decomposition defined in Section 3. Recall that nodes are partitioned into old and new at each timestep t as $\mathcal{V}_t = \mathcal{V}_t^{\text{old}} \cup \mathcal{V}_t^{\text{new}}$, and edges are grouped into four disjoint types: old–old-bank (recurring edges between old nodes), old–old-nobank (new edges between old nodes), old–new (edges connecting old and new nodes), and new–new (edges between new nodes).

The total number of edges at time t is:

$$|\mathcal{E}_t| = |\mathcal{E}_t^{\text{oo-bank}}| + |\mathcal{E}_t^{\text{oo-nobank}}| + |\mathcal{E}_t^{\text{on}}| + |\mathcal{E}_t^{\text{nn}}|.$$

Table 8: Glossary of complexity variables.

Symbol	Meaning
$\mathcal{T}$	Total training epochs, fixed at 100
N_{train}	Number of training snapshots, using the 70% training partition
N_{val} to N	Test set range, using the final 15% partition
L	Number of GCN/GAT layers in the encoder
d	Node embedding dimensionality
H	Predictive heads for edge types $\{\text{o-o}, \text{o-n}, \text{n-n}\}$
$\lvert\mathcal{E}^+_{i,h}\rvert$	Number of positive edge samples per head during training
$\lvert\mathcal{E}^-_{i,h}\rvert$	Number of negative edge samples per head during training
E_t, V_t	Observed edges and nodes in the input context snapshot
$\widehat{\mathcal{E}}, \widehat{\mathcal{V}}$	Target edges and nodes defined by TopER and probability vectors
$\mathcal{B}$	Number of filtration buckets used for node-degree assignment
M	Number of potential edge candidates sampled per node
$s(d)$	Computational cost of the MLP scoring function
$\log \widehat{\mathcal{E}}$	Complexity term for top-K edge selection through sorting

We define the normalized edge-type probabilities as the share of each type in the snapshot:

$$\pi_t^{\text{oo-bank}} = \frac{|\mathcal{E}_t^{\text{oo-bank}}|}{|\mathcal{E}_t|}, \quad \pi_t^{\text{oo-nobank}} = \frac{|\mathcal{E}_t^{\text{oo-nobank}}|}{|\mathcal{E}_t|},$$
$$\pi_t^{\text{on}} = \frac{|\mathcal{E}_t^{\text{on}}|}{|\mathcal{E}_t|}, \quad \pi_t^{\text{nn}} = \frac{|\mathcal{E}_t^{\text{nn}}|}{|\mathcal{E}_t|}.$$

These probabilities quantify the inductive edge-type composition of the current snapshot. For example, $p_{\text{oo}}^{\text{bank}}(t)$ measures the recurrence rate of previously observed edges, while $p_{\text{nn}}(t)$ reflects purely novel interactions among newly arriving nodes.

Algorithm 1: TOPOGED Inference: Budgeted Multi-Phase Snapshot Reconstruction

Input: Observed snapshots $\{\mathcal{G}_{t-k+1}, \ldots, \mathcal{G}_t\}$; Edge Set $\mathcal{B}$; encoder $\Phi(\cdot)$; topology predictor Pred; trunk encoder Enc; type heads $\{M_z\}_{z\in\mathcal{Z}}$.

Output: Predicted snapshot $\widehat{\mathcal{G}}_{t+1} = (\widehat{\mathcal{V}}_{t+1}, \widehat{\mathcal{E}}_{t+1})$ and updated EdgeBank $\mathcal{B}$.

1. Global Budget Forecasting.

Compute $\Phi(\mathcal{G}_\tau)$ for $\tau = t-k+1, \ldots, t$.

Predict $\widehat{\Phi}(\mathcal{G}_{t+1})$, $|\widehat{\mathcal{V}}_{t+1}|$, $\widehat{\mathcal{E}}_{t+1}$, $\widehat{p}^{\text{new}}_{t+1}$ and edge-type proportions $\boldsymbol{\pi}_{t+1}$ using Pred.

Set $\widehat{\mathcal{V}}^{\text{old}}_{t+1} = \lfloor |\widehat{\mathcal{V}}_{t+1}|(1-\widehat{p}^{\text{new}}_{t+1}) \rfloor$ and $\widehat{\mathcal{V}}^{\text{new}}_{t+1} = |\widehat{\mathcal{V}}_{t+1}| - \widehat{\mathcal{V}}^{\text{old}}_{t+1}$.

Set per-type edge budgets $\widehat{c}_z(t+1) = \lfloor \pi_z(t+1)\widehat{\mathcal{E}}_{t+1} \rfloor$ for $z \in \{\text{oo-bank}, \text{oo-nobank}, \text{on}, \text{nn}\}$.

2. Inductive Node Set Assembly.

Reappearing Nodes: Sample $\widehat{N}_{\text{old}}$ nodes from the historical universe $\bigcup_{\tau=t-k+1}^{t} \mathcal{V}_\tau$ using weighted random sampling as in Section 4.3. Initialize $v \in \widehat{\mathcal{V}}^{\text{old}}_{t+1}$ with their most recent observed degree.

Cold-Start Nodes: Initialize $\widehat{N}_{\text{new}}$ node identifiers. Assign each $v \in \widehat{\mathcal{V}}^{\text{new}}_{t+1}$ an initial degree value from the remaining capacity in the predicted descriptor $\widehat{\Phi}(\mathcal{G}_{t+1})$.

Define $\widehat{\mathcal{V}}_{t+1} = \widehat{\mathcal{V}}^{\text{old}}_{t+1} \cup \widehat{\mathcal{V}}^{\text{new}}_{t+1}$.

3. Representation Learning.

Initialize embeddings h: for $v \in \widehat{\mathcal{V}}^{\text{old}}_{t+1}$, use the most recently seen embedding from history; for $v \in \widehat{\mathcal{V}}^{\text{new}}_{t+1}$, initialize with default vectors.

Compute inductive representations $h \leftarrow \text{Enc}(\widehat{\mathcal{V}}_{t+1}, \mathcal{G}_{t-k+1:t})$.

4. Multi-Phase Edge Reconstruction (Top-K Filtering).

Phase I (Bank Recurrence): Define $\mathcal{C}_{\text{oo-bank}} = \{(u,v) \in \mathcal{B} \mid u, v \in \widehat{\mathcal{V}}^{\text{old}}_{t+1}\}$. Add top $\widehat{c}_{\text{oo-bank}}$ pairs scored by $M_{\text{oo-bank}}(h_u, h_v)$ to $\widehat{\mathcal{E}}_{t+1}$.

Phase II (Discovery): Define $\mathcal{C}_{\text{oo-nobank}} = \{(u,v) \in \widehat{\mathcal{V}}^{\text{old}}_{t+1} \times \widehat{\mathcal{V}}^{\text{old}}_{t+1} \mid (u,v) \notin \mathcal{B}\}$. Add top $\widehat{c}_{\text{oo-nobank}}$ pairs scored by $M_{\text{oo-nobank}}$ to $\widehat{\mathcal{E}}_{t+1}$.

Phase III (Inductive Growth): Define $\mathcal{C}_{\text{on}} = \widehat{\mathcal{V}}^{\text{old}}_{t+1} \times \widehat{\mathcal{V}}^{\text{new}}_{t+1}$. Add top $\widehat{c}_{\text{on}}$ pairs scored by M_{on} to $\widehat{\mathcal{E}}_{t+1}$.

Phase IV (New Community): Define $\mathcal{C}_{\text{nn}} = \{(u,v) \in \widehat{\mathcal{V}}^{\text{new}}_{t+1} \times \widehat{\mathcal{V}}^{\text{new}}_{t+1}, u \neq v\}$. Add top $\widehat{c}_{\text{nn}}$ pairs scored by M_{nn} to $\widehat{\mathcal{E}}_{t+1}$.

Note: Embeddings h may optionally be updated via Enc between phases to reflect incremental structural changes.

5. EdgeBank Maintenance.

Update $\mathcal{B} \leftarrow (\mathcal{B} \cup \widehat{\mathcal{E}}_{t+1}) \setminus \mathcal{E}_{expired}$ where $\mathcal{E}_{expired}$ are edges older than k snapshots.

return $\widehat{\mathcal{G}}_{t+1} = (\widehat{\mathcal{V}}_{t+1}, \widehat{\mathcal{E}}_{t+1})$, $\mathcal{B}$.

C.5 Adapting Link-Prediction Baselines to Graph Construction

Benchmark models are based on their ability to predict edge existence. For each potential edge candidate, the probability of edge (u, v), the model outputs a probability p^t_{uv} representing the likelihood of an edge appearing at timestamp t. We learn a global threshold for the dataset and the model to produce a binary adjacency matrix from the probability values. Specifically, an edge is included in the predicted graph if and only if

$$e^{(t)}_{uv} = \begin{cases} 1 & \text{if } p^{(t)}_{uv} \geq T \\ 0 & \text{otherwise} \end{cases}$$

Existing methods primarily perform link prediction by receiving a set of candidate edges, which are a subset of known possible edges in the universe. That is

$$\mathcal{E}_{cand} = \mathcal{E}_{pos} \cup \mathcal{E}_{neg}$$

Where $\mathcal{E}_{pos}$ is the set of "true" edges existing in the current snapshot (the positive candidates) and $\mathcal{E}_{neg}$ is a sampled set of non-existent edges (the negative candidates), where typically $|\mathcal{E}_{neg}| = X \cdot |\mathcal{E}_{pos}|$ for some $X \in \mathbb{Z}^+$.

However, for graph construction, because the set of known nodes in $\widehat{G}_{t+1}$ is unknown, we selected candidate edges as all possible edges between all nodes in the node universe across all timesteps T. That is

$$\mathcal{E}_{cand} = \{(u, v, t) \mid u, v \in V, u \neq v, t \in T\}$$

Edge selection and threshold optimization: . To translate the continuous probability scores from baseline models into a discrete adjacency structure, we use a global decision threshold τ. We optimize τ using the validation partition G_{val} by performing a grid search over the interval $[0, 1]$ with a step size of 0.01. The optimal threshold τ^* is selected to maximize the F1-score across all validation snapshots:

$$\tau^* = \arg\max_{\tau} \mathrm{F1}(\widehat{\mathcal{E}}_\tau, \mathcal{E}_{val})$$

where $\widehat{\mathcal{E}}_\tau = \{e \in \mathcal{E}_{cand} \mid p_e \geq \tau\}$. We then apply this optimized threshold to the test partition to reconstruct the final adjacency matrices.

Bounded Adjacency Reconstruction. To ensure scalability across datasets of varying densities and to prevent memory overflow (OOM) during the construction phase, we implement a heuristic cap on graph growth. For each predicted snapshot $\widehat{\mathcal{G}}_t$, the total number of edges is restricted to at most k_t, defined relative to the cardinality of the preceding snapshot:

$$k_t = 5 \times |\mathcal{E}_{t-1}|$$

If the number of candidate edges exceeding the decision threshold τ surpasses this limit, we perform top-k filtering, selecting only the k_t edges with the highest predicted probabilities. This approach ensures computational tractability while preserving the most structurally significant interactions according to the model's confidence scores.

D Stability Theorem

In this section, we prove Proposition 4.1. A natural question is whether the filtration-based descriptor $\Phi(\mathcal{G}_t)$ is a reliable snapshot state for forecasting: if two snapshots differ by only a few edges, do their descriptors remain close? This is essential for TOPOGED because the topology predictor is trained to forecast $\Phi(\mathcal{G}_{t+1})$ from past descriptors, and its predictions will inevitably carry small errors. If the descriptor were sensitive to minor structural perturbations, these errors could cascade into large reconstruction mistakes in the decoder. The following proposition shows that the degree-sublevel filtration descriptor is stable in a precise sense: editing at most δ edges in a snapshot perturbs the node-count sequence X by at most 2δ in ℓ_1 norm, and perturbs the induced-edge-count sequence Y by an amount controlled by δ and the filtration thresholds. This stability justifies using $\Phi(\mathcal{G}_t)$ as a forecasting target: small prediction errors in descriptor space correspond to small structural deviations in the reconstructed snapshot, making the inverse topology approach principled rather than heuristic.

Proposition 4.1 (Stability under edge edits). *Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ and $\mathcal{G}' = (\mathcal{V}, \mathcal{E}')$ be graphs on the same node set with $|\mathcal{E} \triangle \mathcal{E}'| \leq \delta$. Let X, Y and X', Y' be the degree-sublevel filtration node and induced-edge count sequences computed using thresholds $\epsilon_1 < \cdots < \epsilon_n$.*

(i) If the thresholds are consecutive integers $\{0, 1, \ldots, D\}$, then $\|X - X'\|_1 \leq 2\delta$.

(ii) For any thresholds, $\|Y - Y'\|_1 \leq \delta \sum_{i=1}^{n}(1 + 2\epsilon_i)$.

Proof. Let $\Delta d(v) = \deg_{\mathcal{G}}(v) - \deg_{\mathcal{G}'}(v)$. Each edge in $\mathcal{E} \triangle \mathcal{E}'$ changes the degrees of exactly two endpoints by ± 1, hence

$$\sum_{v \in V} |\Delta d(v)| \leq 2|\mathcal{E} \triangle \mathcal{E}'| \leq 2\delta.$$

Part (i). Assume thresholds are $k \in \{0, 1, \ldots, D\}$ and write $x_k = |\mathcal{V}_k(\mathcal{G})|$ where $\mathcal{V}_k(\mathcal{G}) = \{v : \deg_{\mathcal{G}}(v) \leq k\}$, and similarly x'_k. For each k,

$$x_k - x'_k = \sum_{v \in \mathcal{V}} \Big(\mathbf{1}\{\deg_{\mathcal{G}}(v) \leq k\} - \mathbf{1}\{\deg_{\mathcal{G}'}(v) \leq k\} \Big),$$

so by the triangle inequality,

$$|x_k - x'_k| \le \sum_{v\in\mathcal{V}} \Big|\mathbf{1}\{\deg_{\mathcal{G}}(v) \le k\} - \mathbf{1}\{\deg_{\mathcal{G}'}(v) \le k\}\Big|.$$

Summing over k and swapping the order of summation gives

$$\|X - X'\|_1 = \sum_{k=0}^{D} |x_k - x'_k| \le \sum_{v\in\mathcal{V}} \sum_{k=0}^{D} \Big|\mathbf{1}\{\deg_{\mathcal{G}}(v) \le k\} - \mathbf{1}\{\deg_{\mathcal{G}'}(v) \le k\}\Big|.$$

For a fixed v, the inner sum counts how many integer thresholds lie strictly between $\deg_{\mathcal{G}}(v)$ and $\deg_{\mathcal{G}'}(v)$, which is exactly $|\Delta d(v)|$ whenever $D \ge \max\{\deg_{\mathcal{G}}(v), \deg_{\mathcal{G}'}(v)\}$ and at most $|\Delta d(v)|$ otherwise. Therefore,

$$\|X - X'\|_1 \le \sum_{v\in V} |\Delta d(v)| \le 2\delta.$$

Part (ii). Fix a threshold ϵ_i and let $\mathcal{S} = \mathcal{V}_{\epsilon_i}(\mathcal{G})$ and $\mathcal{S}' = \mathcal{V}_{\epsilon_i}(\mathcal{G}')$. Let $\mathcal{F} = \mathcal{E}\triangle\mathcal{E}'$ denote the set of edited edges and $\mathcal{C} = \mathcal{E} \cap \mathcal{E}'$ the set of common edges. Any edge in the symmetric difference $\mathcal{E}(\mathcal{G}[\mathcal{S}])\triangle\mathcal{E}(\mathcal{G}'[\mathcal{S}'])$ is either an edited edge (in $\mathcal{F}$) or a common edge whose membership in the induced subgraph changes due to a vertex moving between $\mathcal{S}$ and $\mathcal{S}'$. Thus

$$|\mathcal{E}(\mathcal{G}[\mathcal{S}])\triangle\mathcal{E}(\mathcal{G}'[\mathcal{S}'])| \le |\mathcal{F}| + \big|\mathcal{C} \cap \big((\mathcal{S}\times\mathcal{S})\triangle(\mathcal{S}'\times\mathcal{S}')\big)\big|.$$

The first term satisfies $|\mathcal{F}| \le \delta$. For the second term, a common edge changes membership in the induced subgraph only if at least one of its endpoints lies in $\mathcal{S}\triangle\mathcal{S}'$, so

$$\big|\mathcal{C} \cap \big((\mathcal{S}\times\mathcal{S})\triangle(\mathcal{S}'\times\mathcal{S}')\big)\big| \le \sum_{v\in\mathcal{S}\triangle\mathcal{S}'} \deg_{\mathcal{C}}(v),$$

where $\deg_{\mathcal{C}}(v)$ denotes the degree of v in the common-edge graph $(\mathcal{V}, \mathcal{C})$. For any $v \in \mathcal{S}\triangle\mathcal{S}'$, membership differs between $\mathcal{S}$ and $\mathcal{S}'$ only if $\deg_{\mathcal{G}}(v) \ne \deg_{\mathcal{G}'}(v)$, and either $v \in \mathcal{S}$ or $v \in \mathcal{S}'$, so $\min(\deg_{\mathcal{G}}(v), \deg_{\mathcal{G}'}(v)) \le \epsilon_i$. Since $\mathcal{C} \subseteq \mathcal{E}$ and $\mathcal{C} \subseteq \mathcal{E}'$, we have

$$\deg_{\mathcal{C}}(v) \le \min\big(\deg_{\mathcal{G}}(v), \deg_{\mathcal{G}'}(v)\big) \le \epsilon_i.$$

Moreover, $|\mathcal{S}\triangle\mathcal{S}'| \le \sum_{v\in\mathcal{V}} |\Delta d(v)| \le 2\delta$. Combining,

$$\big|\mathcal{C} \cap \big((\mathcal{S}\times\mathcal{S})\triangle(\mathcal{S}'\times\mathcal{S}')\big)\big| \le \epsilon_i\, |\mathcal{S}\triangle\mathcal{S}'| \le 2\delta\epsilon_i.$$

Therefore,

$$|y_i - y'_i| \le |\mathcal{E}(\mathcal{G}[\mathcal{S}])\triangle\mathcal{E}(\mathcal{G}'[\mathcal{S}'])| \le \delta + 2\delta\epsilon_i = \delta(1 + 2\epsilon_i).$$

Summing over all i gives the stated ℓ_1 bound

$$\|Y - Y'\|_1 \le \delta \sum_{i=1}^{n} (1 + 2\epsilon_i). \qquad \square$$

E Experimental Details

This section provides additional experimental details omitted from the main text. Appendix E.1 summarizes the temporal graph baselines, Appendix E.2 defines the graph construction metrics; Appendix E.3 reports the descriptor and budget prediction results used by TopoGED, and Appendix E.4 analyzes computational complexity.

E.1 Baseline Models

In this section, we give further details about the temporal graph learning models we used as a baseline for our work.

TGCN [49] is a combination of GCN and GRU. In particular, GCN is used to learn complex topological structures, while GRU is used to model embeddings dynamically to capture temporal dependence.

HTGN [43] leverages the power of hyperbolic geometry, which is well-suited for capturing hierarchical structures and complex relationships in temporal networks. HTGN maps the temporal graph into hyperbolic space and utilizes hyperbolic graph neural networks and hyperbolic gated recurrent neural networks to model the evolving dynamics. It incorporates two key modules that are hyperbolic temporal contextual self-attention (HTA) and hyperbolic temporal consistency (HTC)-to ensure that temporal dependencies are effectively captured and that the model is both stable and generalizable across various tasks.

GCLSTM [5] combines a Graph Convolutional Network (GCN) and Long Short-Term Memory units to handle both the structural and temporal aspects of evolving networks. The GCN is used to capture the local structural properties of the network at each snapshot, while the LSTM learns the temporal evolution of these snapshots over time.

EvolveGCN [28] is designed to capture the temporal dynamics of graph-structured data. Instead of relying on static node embeddings, EvolveGCN evolves the parameters of a graph convolutional network (GCN) over time. By using a recurrent neural network (RNN) to adapt the GCN parameters, this model is capable of dynamically adjusting during both training and testing. This makes EvolveGCN suitable for evolving graphs with changing observed nodes, although it does not explicitly generate unseen nodes or full future snapshots.

ROLAND [45] is a dynamic graph learning framework that models node representations as hierarchical states, updated recurrently to capture temporal dependencies in evolving graphs. It supports scalable training using techniques like truncated backpropagation through time and meta-learning. In our DTDG setting, we use ROLAND to benchmark its performance and adaptability across diverse temporal networks.

VGRNN [11] introduces a hierarchical variational model for dynamic graphs, extending the variational autoencoder to capture temporal dependencies. The framework utilizes high-level latent random variables within a graph recurrent neural network to model evolving topology and node attributes over time.

E.2 Evaluation Metrics

We evaluate each predicted snapshot $\widehat{\mathcal{G}}_{t+1} = (\widehat{\mathcal{V}}_{t+1}, \widehat{\mathcal{E}}_{t+1})$ against the corresponding ground-truth snapshot $\mathcal{G}_{t+1} = (\mathcal{V}_{t+1}, \mathcal{E}_{t+1})$ over the N test timestamps. Our metrics measure three complementary aspects of forecast quality: structural fidelity, node and edge recovery, and budget errors.

Structural fidelity. For a graph statistic f, we report the mean relative error

$$\mathcal{S}_f = \frac{1}{N} \sum_{t=1}^{N} \left(\frac{f(\widehat{\mathcal{G}}_{t+1}) - f(\mathcal{G}_{t+1})}{f(\mathcal{G}_{t+1}) + \varepsilon} \right),$$

where $\varepsilon > 0$ is a small constant used to avoid division by zero. Lower values indicate closer agreement between the predicted and true snapshots. We apply this evaluation to average node degree, number of unique degrees, average degree centrality, assortativity coefficient, clustering coefficient, density, number of triangles, and the filtration-descriptor error. For the descriptor metric, both graphs are encoded using the degree-based filtration descriptor $\Phi(\cdot)$, and we report the relative error between the resulting descriptor vectors.

- **Avg Node Degree**: Measures the error magnitude in matching the average node connectivity of the ground truth graph.
- **Unique Degree Count**: Evaluates the difference in distribution of distinct degree values across nodes.
- **Degree Centrality**: Tracks differences in centrality distribution errors between true and predicted graphs.
- **Assortativity Coefficient**: Measures the error in mixing patterns (e.g., whether high-degree nodes connect to other high-degree nodes).
- **Clustering Coefficient**: Evaluates local closure and clustering density
- **Density**: Measures the error in overall graph density.

- **Num Triangles**: Tracks the difference in local motif structure by counting triangle formations.
- **Descriptor Norm**: Measures the difference between the predicted structural descriptor and target descriptor through the L2 Norm.
- **Median Extra Nodes**: Measures the median number of over-predicted nodes across all graphs.
- **Median Missing Nodes**: Measures the median number of under-predicted nodes across all graphs.
- **Median Extra Edges**: Measures the median count of over-predicted edges across all graphs.
- **Median Missing Edges**: Measures the median count of under-predicted edges across all graphs.

Node and edge recovery. We evaluate recovery of the true node and edge sets using precision, recall, and F1 scores. For nodes, these metrics compare the predicted active node set $\widehat{\mathcal{V}}_{t+1}$ with the true active node set $\mathcal{V}_{t+1}$. For edges, they compare $\widehat{\mathcal{E}}_{t+1}$ with $\mathcal{E}_{t+1}$, either over all edges or within the inductive edge categories old–old-bank, old–old-nobank, old–new, and new–new. Aggregate scores are computed by averaging the corresponding metric over all test snapshots:

$$\mathcal{S}_f = \frac{1}{N} \sum_{t=1}^{N} f(\widehat{\mathcal{G}}_{t+1}, \mathcal{G}_{t+1}).$$

For these metrics, higher values indicate better recovery.

- **Precision Nodes / Recall Nodes / F1 Nodes**: Evaluates the overall accuracy of predicting the complete active node set through node IDs.
- **Precision Old Nodes / Recall Old Nodes / F1 Old Nodes**: Specifically measures how accurately the model identifies and retains reappearing historical (old) nodes through node IDs.
- **New Nodes Predicted**: Measures the number of new nodes predicted by counting the occurrences in the predicted graph.
- **oo-bank Precision / Recall / F1**: Measures accuracy for old-to-old, previously seen edges through Precision, Recall, and F1 by comparing involved node IDs.
- **oo-nobank Precision / Recall / F1**: Measures accuracy for old-to-old, never seen, edges through Precision, Recall, and F1 by comparing involved node IDs.
- **Edge Precision / Recall / F1**: Measures accuracy for all edge types through Precision, Recall, and F1 by comparing involved node IDs.
- **Num o-n Predicted**: Tracks old-to-new edge predictions.
- **Num n-n Predicted**: Tracks new-to-new edge predictions.

Budget errors. We also report median errors in the predicted node and edge budgets. These metrics quantify over-generation and under-generation separately:

$$\mathrm{ExtraNodes}_t = \max\{|\widehat{\mathcal{V}}_{t+1}| - |\mathcal{V}_{t+1}|, 0\}, \qquad \mathrm{MissingNodes}_t = \max\{|\mathcal{V}_{t+1}| - |\widehat{\mathcal{V}}_{t+1}|, 0\},$$

and

$$\mathrm{ExtraEdges}_t = \max\{|\widehat{\mathcal{E}}_{t+1}| - |\mathcal{E}_{t+1}|, 0\}, \qquad \mathrm{MissingEdges}_t = \max\{|\mathcal{E}_{t+1}| - |\widehat{\mathcal{E}}_{t+1}|, 0\}.$$

We summarize each quantity by its median over test snapshots:

$$\mathcal{S}_f = \mathrm{median}\left(\{f_t\}_{t=1}^{N}\right).$$

Lower values indicate better calibration of the predicted graph size. Together, these metrics assess whether a model reconstructs the correct global structure, recovers the correct active nodes and edges, and avoids excessive over- or under-generation.

E.3 Descriptor Prediction Performance

To predict the descriptor and probability vectors used to guide graph construction, we tested multiple prediction methods. We tested the models SMA [47], VAR [36], V-EWMA [2], SSM [17], and VECM [16].

The performance is detailed in Tables 9 and 10. As the tables show, V-EWMA has the best average performance, achieving the lowest mean error for both node and edge prediction, with average errors of 8.12% and 10.82%, respectively. This indicates that V-EWMA provides the most accurate joint prediction of the descriptor and probability vectors used in graph construction.

Table 9: Percent Error Nodes

Dataset	**SMA**	**VAR**	**V-EWMA**	**SSM**	**VECM**
Collegemsg	15.10	23.43	**11.64**	27.88	18.94
Mathoverflow	**0.28**	-1.27	0.41	-3.65	-1.76
Adex	**7.41**	60.63	9.84	11.23	44.21
Aeternity	-1.64	17.88	0.85	3.42	**-0.81**
Aion	**0.02**	10.08	-0.27	22.31	13.67
Aragon	6.69	47.72	**5.51**	18.55	39.06
Bancor	**0.28**	46.64	1.31	5.42	27.22
Centra	43.65	63.00	**37.33**	64.64	57.76
Cindicator	5.85	60.88	**4.52**	40.49	48.90
Coindash	**1.89**	119.84	8.42	15.36	117.00
Dgd	14.15	**4.95**	10.85	22.94	9.20
Iconomi	17.30	26.76	**15.62**	34.18	24.04
Reddit_b	0.84	-3.24	0.67	**0.65**	-3.86
Tgbl-wiki	9.13	**5.32**	6.96	7.26	6.44
Average	8.64	34.47	**8.12**	19.33	28.57

Table 10: Percent Error Edges

Dataset	**SMA**	**VAR**	**V-EWMA**	**SSM**	**VECM**
Collegemsg	29.06	41.57	23.65	46.19	**20.99**
Mathoverflow	**1.11**	-1.25	1.23	-3.22	-1.38
Adex	**12.25**	71.53	16.57	14.23	55.26
Aeternity	-2.21	13.14	**-0.01**	2.02	-2.08
Aion	0.75	7.79	**0.49**	19.71	10.93
Aragon	7.48	52.68	**6.25**	19.80	43.73
Bancor	**1.50**	45.89	2.93	6.45	27.41
Centra	51.01	56.46	**41.90**	80.14	65.42
Cindicator	6.42	48.65	**5.11**	40.96	37.83
Coindash	**3.67**	103.73	10.09	18.76	112.69
Dgd	18.87	**9.40**	14.79	29.81	13.72
Iconomi	20.77	26.55	**18.55**	37.92	23.83
Reddit_b	2.64	-4.84	**2.10**	2.70	-5.02
Tgbl-wiki	10.55	**6.88**	7.88	8.42	9.59
Average	11.70	34.16	**10.82**	23.13	29.49

E.4 Computational Complexity Analysis

Let $n_t = |\mathcal{V}_t|$ and $m_t = |\mathcal{E}_t|$ denote the number of nodes and edges in the observed snapshot, and let $\widehat{n}_{t+1} = |\widehat{\mathcal{V}}_{t+1}|$ and $\widehat{m}_{t+1} = |\widehat{\mathcal{E}}_{t+1}|$ denote the predicted node and edge budgets. Let d be the

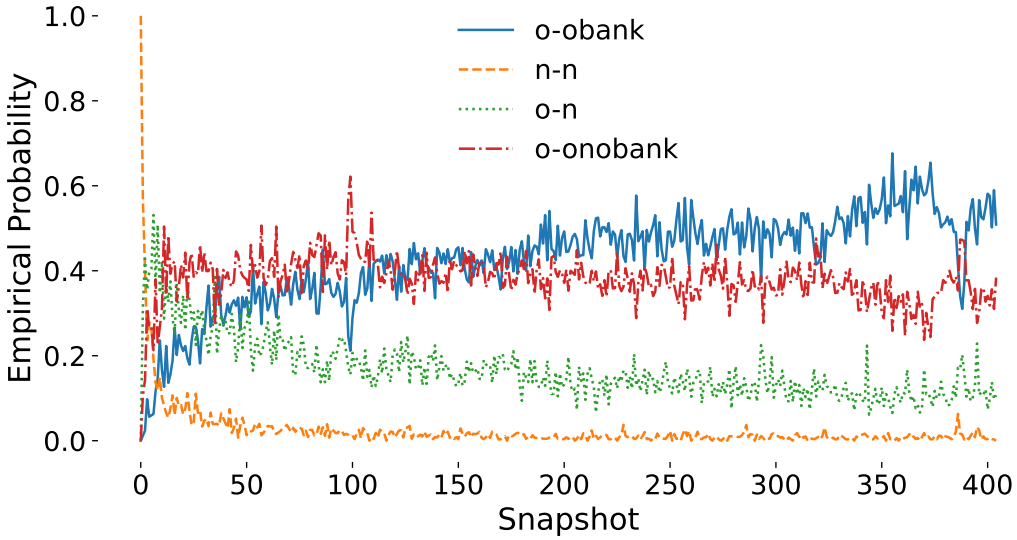


Figure 9: Empirical probabilities of edge types over time for the Reddit-B network.

embedding dimension, L the number of GCN layers, and M the number of sampled candidate partners per node.

The filtration encoder computes degrees and cumulative node and edge counts over degree thresholds, which costs $O(n_t + m_t)$ per snapshot once the threshold set is fixed. The descriptor predictor operates on a fixed-length sequence of filtration vectors, so its cost does not scale with the graph size. The memory module scans the last k snapshots to recover candidate old nodes and compute reappearance scores, with worst-case cost $O(\sum_{\tau=t-k}^{t} m_\tau)$. This avoids scoring the full historical node universe.

The decoder is the dominant term. Each decoding phase scores a restricted candidate set instead of all possible node pairs. With at most M candidate partners per predicted node, candidate scoring costs $O(M\widehat{n}_{t+1}s(d))$, where $s(d)$ is the cost of scoring one candidate edge. Selecting the top edges adds $O(M\widehat{n}_{t+1}\log \widehat{m}_{t+1})$. Recomputing embeddings with an L-layer sparse GCN over the partially generated graph costs $O(L(\widehat{m}_{t+1}d + \widehat{n}_{t+1}d^2))$ across a constant number of decoding phases.

Thus, the per-snapshot inference cost is $O(n_t + m_t + \sum_{\tau=t-k}^{t} m_\tau + M\widehat{n}_{t+1}(s(d) + \log \widehat{m}_{t+1}) + L(\widehat{m}_{t+1}d + \widehat{n}_{t+1}d^2))$. For fixed k, M, L, and descriptor length, the cost scales near-linearly in the observed memory window and the predicted snapshot size. Training has the same leading terms, with the edge-scoring cost multiplied by $(1 + r)$ when r negative edges are sampled per positive edge.

F Results By Dataset

In this section, we present a granular performance analysis across all evaluated datasets to contextualize the macro averages provided in Table 3. TopoGED consistently demonstrates superior predictive capacity, with the most significant gains occurring in `F1 Nodes`, `Edge F1`, and `F1 Old Nodes`.

Figure 9 visualizes why fixed edge priors fail on temporal datasets. The empirical mix of o-obank, o-onobank, o-n, and n-n edges shifts over time, so a model that treats edge formation as stationary will allocate probability mass to the wrong edge types during regime changes.

Notably, while most baseline models struggle to maintain non-zero scores in edge-level forecasting, TopoGED maintains a robust performance margin. Furthermore, the model exhibits substantial precision in capturing graph topology, as evidenced by the `Structure Evaluation` metrics. On average, TopoGED outperforms the strongest baseline in 60% of graph property tasks, frequently achieving results orders of magnitude closer to the ground truth than the nearest competing model.

Detailed results for all datasets are reported in Tables 11–24, covering CollegeMessage, MathOverflow, Adex, Aeternity, Aion, Aragon, Bancor, Centra, Cindicator, Coindash, DGD, Iconomi, Reddit_B, and TGBL-Wiki.

Table 11: Comprehensive evaluation results—including edge, node, and structural metrics—for all benchmark models on **CollegeMsg**. $\rightarrow 0$ indicates that values closer to 0 are better. $\uparrow$ denotes that higher values are better, while $\downarrow$ denotes that lower values are better. **Bold** indicates best, the second-best is underlined.

	Metric	HTGN	ROLAND	VGRNN	TGCN	GCLSTM	EvolveGCN	TopoGED
Edge Evaluation	oo-bank Precision↑	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	$\mathbf{0.26 \pm 0.12}$
	oo-bank Recall↑	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	$\mathbf{0.45 \pm 0.21}$
	oo-bank F1↑	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	$\mathbf{0.32 \pm 0.12}$
	oo-nobank Precision↑	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	$\mathbf{0.01 \pm 0.03}$
	oo-nobank Recall↑	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	$\mathbf{0.01 \pm 0.05}$
	oo-nobank F1↑	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	$\mathbf{0.01 \pm 0.04}$
	Edge Precision↑	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	$\mathbf{0.16 \pm 0.07}$
	Edge Recall↑	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	$\mathbf{0.25 \pm 0.13}$
	Edge F1↑	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	$\mathbf{0.19 \pm 0.07}$
	Num o-n Predicted $\rightarrow 0$	1.57 ± 2.62	0.54 ± 2.22	$\underline{-0.16 \pm 2.00}$	1.75 ± 8.89	$\mathbf{0.15 \pm 2.07}$	1.15 ± 3.13	0.96 ± 2.42
	Num n-n Predicted $\rightarrow 0$	$\mathbf{-0.04 \pm 0.19}$	$\mathbf{-0.04 \pm 0.19}$	$\mathbf{-0.04 \pm 0.19}$	0.14 ± 0.71	$\mathbf{-0.04 \pm 0.19}$	0.09 ± 0.33	$\underline{0.07 \pm 0.60}$
Node Evaluation	Precision Nodes↑	0.02 ± 0.02	0.02 ± 0.02	$\underline{0.03 \pm 0.04}$	0.01 ± 0.02	0.02 ± 0.02	0.02 ± 0.02	$\mathbf{0.43 \pm 0.11}$
	Recall Nodes↑	0.06 ± 0.04	0.02 ± 0.02	0.02 ± 0.02	0.01 ± 0.02	0.01 ± 0.01	$\underline{0.09 \pm 0.05}$	$\mathbf{0.46 \pm 0.13}$
	F1 Nodes↑	0.03 ± 0.02	0.02 ± 0.02	0.02 ± 0.03	0.01 ± 0.02	0.01 ± 0.02	$\underline{0.04 \pm 0.02}$	$\mathbf{0.44 \pm 0.10}$
	Precision Old Nodes↑	0.02 ± 0.02	0.02 ± 0.02	$\underline{0.03 \pm 0.04}$	0.00 ± 0.01	0.02 ± 0.02	0.02 ± 0.01	$\mathbf{0.43 \pm 0.11}$
	Recall Old Nodes↑	0.06 ± 0.04	0.02 ± 0.02	0.02 ± 0.03	0.00 ± 0.01	0.01 ± 0.02	$\underline{0.09 \pm 0.05}$	$\mathbf{0.46 \pm 0.13}$
	F1 Old Nodes↑	$\underline{0.04 \pm 0.02}$	0.02 ± 0.02	0.02 ± 0.03	0.00 ± 0.01	0.01 ± 0.02	$\underline{0.04 \pm 0.02}$	$\mathbf{0.43 \pm 0.10}$
	New Nodes Predicted↑	$\mathbf{-0.15 \pm 0.64}$	$\underline{-0.30 \pm 0.71}$	-0.57 ± 0.67	-0.31 ± 1.58	-0.53 ± 0.72	1.03 ± 2.96	0.42 ± 1.41
Structure Evaluation	Avg Node Degree $\rightarrow 0$	0.37 ± 0.21	2.59 ± 1.19	4.75 ± 1.60	3.18 ± 0.73	4.70 ± 1.69	$\mathbf{-0.05 \pm 0.10}$	$\underline{0.36 \pm 0.19}$
	Unique Degree Count$\rightarrow 0$	0.72 ± 0.55	2.20 ± 1.10	1.89 ± 1.03	2.52 ± 0.97	2.44 ± 1.03	$\mathbf{0.37 \pm 0.47}$	$\underline{0.64 \pm 0.48}$
	Degree Centrality$\rightarrow 0$	$\underline{-0.45 \pm 0.11}$	3.02 ± 2.58	9.07 ± 3.84	4.20 ± 1.01	9.07 ± 4.30	-0.73 ± 0.07	$\mathbf{0.28 \pm 0.24}$
	Assortativity Coefficient$\rightarrow 0$	$\underline{-5.19 \pm 6.47}$	6.98 ± 17.90	6.02 ± 14.07	5.45 ± 13.81	5.49 ± 13.43	-8.03 ± 12.29	$\mathbf{1.13 \pm 7.92}$
	Clustering Coefficient$\rightarrow 0$	$\underline{0.39 \pm 1.03}$	1.24 ± 3.92	1.64 ± 4.80	1.33 ± 3.75	1.52 ± 4.12	$\underline{0.20 \pm 0.48}$	$\mathbf{0.07 \pm 0.06}$
	Density$\rightarrow 0$	$\underline{-0.45 \pm 0.11}$	3.02 ± 2.58	9.07 ± 3.84	4.20 ± 1.01	9.07 ± 4.30	-0.73 ± 0.07	$\mathbf{0.28 \pm 0.24}$
	Num Triangles$\rightarrow 0$	$\underline{22.46 \pm 16.69}$	130.32 ± 100.27	214.75 ± 143.20	147.96 ± 99.07	219.50 ± 165.61	$\underline{18.43 \pm 22.63}$	$\mathbf{3.11 \pm 3.20}$
	Descriptor Norm$\rightarrow 0$	157.41 ± 64.71	91.78 ± 36.40	103.52 ± 45.81	$\underline{86.98 \pm 27.94}$	100.15 ± 37.70	318.29 ± 120.85	$\mathbf{35.42 \pm 11.91}$
	Median Extra Nodes$\rightarrow 0$	57.00 ± 6.14	0.00 ± 1.34	$\mathbf{0.00 \pm 0.00}$	$\mathbf{0.00 \pm 0.00}$	$\mathbf{0.00 \pm 0.00}$	93.50 ± 10.56	$\underline{0.00 \pm 1.19}$
	Median Missing Nodes$\rightarrow 0$	$\mathbf{0.00 \pm 0.00}$	2.50 ± 2.22	15.00 ± 1.34	9.00 ± 0.99	15.00 ± 2.19	$\mathbf{0.00 \pm 0.00}$	$\underline{0.50 \pm 1.19}$
	Median Extra Edges$\rightarrow 0$	57.00 ± 8.15	63.50 ± 8.08	$\underline{57.00 \pm 8.11}$	57.00 ± 8.39	57.00 ± 8.39	57.00 ± 8.20	$\mathbf{7.50 \pm 2.67}$
	Median Missing Edges$\rightarrow 0$	$\mathbf{0.00 \pm 0.00}$	$\mathbf{0.00 \pm 0.00}$	$\mathbf{0.00 \pm 0.00}$	$\mathbf{0.00 \pm 0.00}$	$\mathbf{0.00 \pm 0.00}$	$\mathbf{0.00 \pm 0.00}$	$\mathbf{0.00 \pm 0.00}$

Table 12: Comprehensive evaluation results—including edge, node, and structural metrics—for all benchmark models on **MathOverflow**. $\rightarrow 0$ indicates that values closer to 0 are better. $\uparrow$ denotes that higher values are better, while $\downarrow$ denotes that lower values are better. **Bold** indicates best, the second-best is underlined.

	Metric	HTGN	ROLAND	VGRNN	TGCN	GCLSTM	EvolveGCN	TopoGED
Edge Evaluation	oo-bank Precision↑	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	$\mathbf{0.07 \pm 0.03}$
	oo-bank Recall↑	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	$\mathbf{0.09 \pm 0.04}$
	oo-bank F1↑	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	$\mathbf{0.08 \pm 0.04}$
	oo-nobank Precision↑	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	$\mathbf{0.00 \pm 0.01}$
	oo-nobank Recall↑	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	$\mathbf{0.00 \pm 0.01}$
	oo-nobank F1↑	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	$\mathbf{0.00 \pm 0.01}$
	Edge Precision↑	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	$\mathbf{0.02 \pm 0.01}$
	Edge Recall↑	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	$\mathbf{0.03 \pm 0.01}$
	Edge F1↑	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	$\mathbf{0.02 \pm 0.01}$
	Num o-n Predicted$\rightarrow 0$	1.67 ± 1.16	1.99 ± 1.45	2.56 ± 1.91	$\underline{-0.36 \pm 0.31}$	3.57 ± 2.53	2.48 ± 1.69	$\mathbf{0.14 \pm 0.40}$
	Num n-n Predicted$\rightarrow 0$	85.49 ± 76.27	82.35 ± 79.78	76.85 ± 68.83	$\underline{98.11 \pm 87.26}$	$\underline{69.71 \pm 61.45}$	79.39 ± 75.28	$\mathbf{0.45 \pm 1.63}$
Node Evaluation	Precision Nodes↑	0.01 ± 0.00	0.01 ± 0.01	0.01 ± 0.02	0.01 ± 0.01	$\underline{0.02 \pm 0.02}$	0.01 ± 0.00	$\mathbf{0.42 \pm 0.06}$
	Recall Nodes↑	0.01 ± 0.01	0.01 ± 0.01	0.00 ± 0.01	0.00 ± 0.01	0.01 ± 0.01	$\underline{0.02 \pm 0.01}$	$\mathbf{0.41 \pm 0.04}$
	F1 Nodes↑	$\underline{0.01 \pm 0.01}$	$\underline{0.01 \pm 0.01}$	$\underline{0.01 \pm 0.01}$	$\underline{0.01 \pm 0.01}$	$\underline{0.01 \pm 0.01}$	0.01 ± 0.00	$\mathbf{0.41 \pm 0.04}$
	Precision Old Nodes↑	0.05 ± 0.04	$\underline{0.09 \pm 0.07}$	0.11 ± 0.20	$\underline{0.18 \pm 0.23}$	0.09 ± 0.08	0.05 ± 0.03	$\mathbf{0.36 \pm 0.06}$
	Recall Old Nodes↑	0.01 ± 0.01	0.01 ± 0.01	0.01 ± 0.01	0.00 ± 0.01	0.01 ± 0.01	$\underline{0.02 \pm 0.01}$	$\mathbf{0.35 \pm 0.04}$
	F1 Old Nodes↑	0.02 ± 0.01	0.02 ± 0.01	0.01 ± 0.01	0.01 ± 0.01	0.02 ± 0.01	$\underline{0.03 \pm 0.01}$	$\mathbf{0.35 \pm 0.04}$
	New Nodes Predicted↑	10.94 ± 5.47	4.75 ± 2.16	$\underline{1.61 \pm 1.21}$	2.61 ± 1.61	2.26 ± 1.64	17.13 ± 8.81	$\mathbf{0.07 \pm 0.51}$
Structure Evaluation	Avg Node Degree$\rightarrow 0$	0.73 ± 0.15	2.47 ± 0.80	6.79 ± 2.14	4.92 ± 0.48	4.82 ± 1.08	$\underline{0.13 \pm 0.15}$	$\mathbf{0.07 \pm 0.06}$
	Unique Degree Count$\rightarrow 0$	$\underline{0.80 \pm 0.28}$	2.22 ± 0.57	1.65 ± 0.73	2.72 ± 0.58	2.65 ± 0.63	1.52 ± 0.41	$\mathbf{0.46 \pm 0.27}$
	Degree Centrality$\rightarrow 0$	$\underline{0.11 \pm 0.17}$	3.79 ± 2.31	23.63 ± 14.41	12.46 ± 2.13	12.11 ± 4.04	-0.51 ± 0.16	$\mathbf{0.09 \pm 0.17}$
	Assortativity Coefficient$\rightarrow 0$	-188.57 ± 1073.62	141.27 ± 811.37	144.25 ± 825.39	127.30 ± 740.41	147.56 ± 851.57	$\underline{-54.55 \pm 314.30}$	$\mathbf{51.44 \pm 300.12}$
	Clustering Coefficient$\rightarrow 0$	13.17 ± 9.39	18.98 ± 14.78	30.79 ± 23.61	29.39 ± 20.61	26.02 ± 18.98	$\underline{6.50 \pm 5.57}$	$\mathbf{2.45 \pm 3.26}$
	Density $\rightarrow 0$	$\underline{0.11 \pm 0.17}$	3.79 ± 2.31	23.63 ± 14.41	12.46 ± 2.13	12.11 ± 4.04	-0.51 ± 0.16	$\mathbf{0.09 \pm 0.17}$
	Num Triangles$\rightarrow 0$	$\underline{174.69 \pm 103.80}$	418.85 ± 271.60	972.69 ± 616.75	646.03 ± 386.53	707.80 ± 425.57	209.57 ± 153.77	$\mathbf{6.45 \pm 6.70}$
	Descriptor Norm$\rightarrow 0$	$\underline{349.89 \pm 72.70}$	$\underline{287.23 \pm 48.77}$	378.51 ± 103.85	330.13 ± 45.08	329.96 ± 49.60	626.17 ± 167.32	$\mathbf{98.76 \pm 25.30}$
	Median Extra Nodes$\rightarrow 0$	80.00 ± 4.28	$\mathbf{0.00 \pm 0.00}$	$\mathbf{0.00 \pm 0.00}$	$\mathbf{0.00 \pm 0.00}$	$\mathbf{0.00 \pm 0.00}$	206.00 ± 14.61	$\underline{0.00 \pm 1.87}$
	Median Missing Nodes$\rightarrow 0$	$\mathbf{0.00 \pm 0.00}$	25.00 ± 6.56	91.00 ± 7.21	78.00 ± 4.21	80.00 ± 4.70	$\mathbf{0.00 \pm 0.00}$	$\underline{3.00 \pm 3.53}$
	Median Extra Edges$\rightarrow 0$	230.00 ± 14.01	$\underline{209.00 \pm 23.18}$	227.00 ± 16.99	227.00 ± 15.30	227.00 ± 16.16	227.00 ± 16.05	$\mathbf{8.00 \pm 4.38}$
	Median Missing Edges$\rightarrow 0$	$\mathbf{0.00 \pm 0.00}$	$\mathbf{0.00 \pm 0.00}$	$\mathbf{0.00 \pm 0.00}$	$\mathbf{0.00 \pm 0.00}$	$\mathbf{0.00 \pm 0.00}$	$\mathbf{0.00 \pm 0.00}$	$\underline{0.00 \pm 2.34}$

Table 13: Comprehensive evaluation results—including edge, node, and structural metrics—for all benchmark models on **Adex**. $\rightarrow 0$ indicates that values closer to 0 are better. $\uparrow$ denotes that higher values are better, while $\downarrow$ denotes that lower values are better. **Bold** indicates best, the second-best is underlined.

	Metric	HTGN	ROLAND	VGRNN	TGCN	GCLSTM	EvolveGCN	TopoGED
Edge Evaluation	oo-bank Precision↑	0.01 ± 0.05	0.00 ± 0.00	0.09 ± 0.21	0.00 ± 0.00	$\mathbf{0.25 \pm 0.23}$	0.03 ± 0.15	0.20 ± 0.12
	oo-bank Recall↑	0.00 ± 0.00	0.00 ± 0.00	0.01 ± 0.03	0.00 ± 0.00	0.06 ± 0.06	0.00 ± 0.02	$\mathbf{0.27 \pm 0.13}$
	oo-bank F1↑	0.00 ± 0.01	0.00 ± 0.00	0.02 ± 0.05	0.00 ± 0.00	0.09 ± 0.08	0.00 ± 0.03	$\mathbf{0.22 \pm 0.10}$
	oo-nobank Precision↑	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	$\mathbf{0.01 \pm 0.03}$
	oo-nobank Recall↑	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	$\mathbf{0.01 \pm 0.02}$	0.00 ± 0.00	$\mathbf{0.01 \pm 0.03}$
	oo-nobank F1↑	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	$\mathbf{0.01 \pm 0.03}$
	Edge Precision↑	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	$\mathbf{0.10 \pm 0.05}$
	Edge Recall↑	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.01	0.00 ± 0.00	0.02 ± 0.02	0.00 ± 0.01	$\mathbf{0.09 \pm 0.05}$
	Edge F1↑	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.01 ± 0.00	0.00 ± 0.00	$\mathbf{0.09 \pm 0.04}$
	Num o-n Predicted→ 0	-0.88 ± 0.10	3.41 ± 1.97	2.55 ± 2.19	-0.91 ± 0.10	4.05 ± 3.35	0.29 ± 2.92	$\mathbf{0.19 \pm 0.82}$
	Num n-n Predicted→ 0	$\mathbf{-0.50 \pm 0.68}$	57.73 ± 45.97	43.89 ± 45.76	-0.89 ± 0.32	24.11 ± 23.73	21.25 ± 52.82	0.62 ± 1.69
Node Evaluation	Precision Nodes↑	0.00 ± 0.00	0.04 ± 0.03	0.05 ± 0.03	0.00 ± 0.00	0.08 ± 0.02	0.01 ± 0.01	$\mathbf{0.55 \pm 0.12}$
	Recall Nodes↑	0.00 ± 0.01	0.02 ± 0.01	0.02 ± 0.01	0.00 ± 0.00	0.10 ± 0.03	0.03 ± 0.05	$\mathbf{0.56 \pm 0.11}$
	F1 Nodes↑	0.00 ± 0.00	0.03 ± 0.01	0.03 ± 0.01	0.00 ± 0.00	0.09 ± 0.02	0.01 ± 0.02	$\mathbf{0.54 \pm 0.09}$
	Precision Old Nodes↑	0.00 ± 0.00	0.07 ± 0.03	0.10 ± 0.07	0.00 ± 0.00	0.14 ± 0.04	0.01 ± 0.03	$\mathbf{0.31 \pm 0.09}$
	Recall Old Nodes↑	0.00 ± 0.01	0.03 ± 0.02	0.04 ± 0.02	0.00 ± 0.00	0.17 ± 0.05	0.04 ± 0.07	$\mathbf{0.32 \pm 0.07}$
	F1 Old Nodes↑	0.00 ± 0.00	0.04 ± 0.02	0.05 ± 0.02	0.00 ± 0.00	0.15 ± 0.03	0.02 ± 0.04	$\mathbf{0.30 \pm 0.05}$
	New Nodes Predicted↑	-0.85 ± 0.12	-0.33 ± 0.35	-0.48 ± 0.29	-0.98 ± 0.01	0.41 ± 0.46	$\mathbf{0.08 \pm 1.82}$	0.12 ± 0.59
Structure Evaluation	Avg Node Degree→ 0	-0.40 ± 0.08	7.91 ± 2.93	7.59 ± 3.32	4.46 ± 1.44	3.26 ± 3.56	-0.12 ± 0.39	$\mathbf{0.02 \pm 0.15}$
	Unique Degree Count→ 0	-0.60 ± 0.11	1.11 ± 1.04	0.61 ± 0.56	1.87 ± 0.78	0.83 ± 0.95	$\mathbf{-0.06 \pm 0.65}$	0.09 ± 0.28
	Degree Centrality→ 0	-0.85 ± 0.05	21.15 ± 22.28	21.76 ± 14.62	9.77 ± 3.67	3.10 ± 5.80	-0.71 ± 0.15	$\mathbf{0.05 \pm 0.43}$
	Assortativity Coefficient→ 0	-2.58 ± 0.89	1.29 ± 0.93	0.99 ± 0.67	1.19 ± 0.60	1.30 ± 0.67	-1.33 ± 1.27	$\mathbf{0.88 \pm 0.56}$
	Clustering Coefficient→ 0	$\mathbf{0.37 \pm 1.45}$	38.29 ± 72.55	34.53 ± 65.81	35.92 ± 67.31	31.98 ± 61.46	6.35 ± 13.46	1.54 ± 4.16
	Density→ 0	-0.85 ± 0.05	21.15 ± 22.28	21.76 ± 14.62	9.77 ± 3.67	3.10 ± 5.80	-0.71 ± 0.15	$\mathbf{0.05 \pm 0.43}$
	Num Triangles→ 0	$\mathbf{2.92 \pm 11.49}$	2182.74 ± 6414.98	1277.94 ± 1705.64	755.62 ± 1227.33	895.09 ± 2437.47	99.45 ± 498.74	5.06 ± 7.96
	Descriptor Norm→ 0	1479.50 ± 1422.56	501.81 ± 485.32	470.69 ± 486.77	351.54 ± 442.18	495.11 ± 627.77	754.83 ± 441.84	$\mathbf{180.45 \pm 279.10}$
	Median Extra Nodes→ 0	314.00 ± 32.35	$\mathbf{0.00 \pm 0.00}$	$\mathbf{0.00 \pm 0.00}$	$\mathbf{0.00 \pm 0.00}$	36.00 ± 9.65	211.00 ± 14.68	4.00 ± 3.93
	Median Missing Nodes→ 0	$\mathbf{0.00 \pm 0.00}$	39.00 ± 5.65	51.00 ± 6.32	41.00 ± 6.52	$\mathbf{0.00 \pm 0.00}$	$\mathbf{0.00 \pm 0.00}$	0.00 ± 1.91
	Median Extra Edges→ 0	132.00 ± 14.89	294.00 ± 25.96	203.00 ± 19.02	148.00 ± 19.28	308.00 ± 29.51	129.00 ± 16.17	$\mathbf{1.00 \pm 4.89}$
	Median Missing Edges→ 0	$\mathbf{0.00 \pm 0.00}$	$\mathbf{0.00 \pm 0.00}$	$\mathbf{0.00 \pm 0.00}$	$\mathbf{0.00 \pm 0.00}$	$\mathbf{0.00 \pm 0.00}$	$\mathbf{0.00 \pm 0.00}$	0.00 ± 2.50

Table 14: Comprehensive evaluation results—including edge, node, and structural metrics—for all benchmark models on **Aeternity**. $\rightarrow 0$ indicates that values closer to 0 are better. $\uparrow$ denotes that higher values are better, while $\downarrow$ denotes that lower values are better. **Bold** indicates best, the second-best is underlined.

	Metric	HTGN	ROLAND	VGRNN	TGCN	GCLSTM	EvolveGCN	TopoGED
Edge Evaluation	oo-bank Precision↑	0.00 ± 0.00	0.08 ± 0.28	0.03 ± 0.17	0.00 ± 0.00	0.06 ± 0.23	0.06 ± 0.17	$\mathbf{0.20 \pm 0.07}$
	oo-bank Recall↑	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.01	$\mathbf{0.28 \pm 0.13}$
	oo-bank F1↑	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.01	$\mathbf{0.22 \pm 0.07}$
	oo-nobank Precision↑	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.02	$\mathbf{0.01 \pm 0.02}$
	oo-nobank Recall↑	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	$\mathbf{0.01 \pm 0.03}$
	oo-nobank F1↑	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	$\mathbf{0.01 \pm 0.02}$
	Edge Precision↑	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	$\mathbf{0.11 \pm 0.04}$
	Edge Recall↑	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	$\mathbf{0.11 \pm 0.05}$
	Edge F1↑	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	$\mathbf{0.11 \pm 0.04}$
	Num o-n Predicted→ 0	-0.81 ± 0.12	-0.10 ± 0.44	0.68 ± 1.02	-0.02 ± 2.52	0.84 ± 1.47	0.31 ± 2.05	$\mathbf{0.00 \pm 0.33}$
	Num n-n Predicted→ 0	128.35 ± 75.97	105.26 ± 58.75	196.59 ± 115.27	114.37 ± 83.36	214.64 ± 131.71	233.41 ± 141.90	$\mathbf{0.31 \pm 1.06}$
Node Evaluation	Precision Nodes↑	0.00 ± 0.00	0.01 ± 0.01	0.01 ± 0.01	0.00 ± 0.00	0.01 ± 0.00	0.01 ± 0.00	$\mathbf{0.48 \pm 0.11}$
	Recall Nodes↑	0.00 ± 0.00	0.01 ± 0.01	0.00 ± 0.00	0.00 ± 0.00	0.01 ± 0.01	0.02 ± 0.01	$\mathbf{0.47 \pm 0.14}$
	F1 Nodes↑	0.00 ± 0.00	0.01 ± 0.01	0.01 ± 0.00	0.00 ± 0.00	0.01 ± 0.00	0.01 ± 0.00	$\mathbf{0.46 \pm 0.11}$
	Precision Old Nodes↑	0.06 ± 0.05	0.29 ± 0.17	0.37 ± 0.33	0.02 ± 0.06	0.34 ± 0.11	$\mathbf{0.41 \pm 0.12}$	0.30 ± 0.09
	Recall Old Nodes↑	0.01 ± 0.01	0.01 ± 0.01	0.01 ± 0.00	0.00 ± 0.00	0.02 ± 0.01	0.03 ± 0.02	$\mathbf{0.29 \pm 0.10}$
	F1 Old Nodes↑	0.01 ± 0.01	0.03 ± 0.02	0.01 ± 0.01	0.00 ± 0.01	0.03 ± 0.02	0.05 ± 0.03	$\mathbf{0.29 \pm 0.08}$
	New Nodes Predicted↑	11.32 ± 4.69	1.09 ± 0.93	0.59 ± 0.73	2.29 ± 2.83	3.72 ± 1.71	6.18 ± 3.06	$\mathbf{0.00 \pm 0.32}$
Structure Evaluation	Avg Node Degree→ 0	-0.26 ± 0.43	2.94 ± 1.26	8.71 ± 3.73	2.68 ± 1.89	2.75 ± 2.18	2.08 ± 3.19	$\mathbf{0.00 \pm 0.08}$
	Unique Degree Count→ 0	-0.33 ± 0.38	1.27 ± 0.58	0.64 ± 0.56	1.23 ± 0.97	1.41 ± 1.21	-0.14 ± 0.85	$\mathbf{-0.06 \pm 0.24}$
	Degree Centrality→ 0	-0.69 ± 0.50	6.50 ± 4.77	30.12 ± 56.24	5.55 ± 4.72	3.52 ± 7.69	3.44 ± 11.12	$\mathbf{0.10 \pm 0.44}$
	Assortativity Coefficient→ 0	-2.42 ± 0.34	0.74 ± 0.42	1.10 ± 0.36	0.80 ± 0.45	0.71 ± 0.50	1.46 ± 0.53	$\mathbf{0.64 \pm 0.31}$
	Clustering Coefficient→ 0	8.37 ± 20.58	87.08 ± 195.38	105.80 ± 206.62	51.98 ± 99.18	88.01 ± 196.81	87.57 ± 194.23	$\mathbf{6.70 \pm 14.43}$
	Density→ 0	-0.69 ± 0.50	6.50 ± 4.77	30.12 ± 56.24	5.55 ± 4.72	3.52 ± 7.69	3.44 ± 11.12	$\mathbf{0.10 \pm 0.44}$
	Num Triangles→ 0	112.86 ± 289.18	1235.52 ± 1708.21	4023.98 ± 7267.32	832.73 ± 871.23	1655.07 ± 3540.89	1395.16 ± 4023.10	$\mathbf{10.21 \pm 14.78}$
	Descriptor Norm→ 0	2185.37 ± 1115.69	666.41 ± 657.85	1289.74 ± 1286.05	754.89 ± 728.87	1272.95 ± 1046.26	1567.32 ± 1128.84	$\mathbf{384.54 \pm 599.03}$
	Median Extra Nodes→ 0	625.00 ± 25.84	$\mathbf{0.00 \pm 0.00}$	$\mathbf{0.00 \pm 0.00}$	$\mathbf{0.00 \pm 4.27}$	138.00 ± 16.87	322.50 ± 19.87	1.00 ± 5.34
	Median Missing Nodes→ 0	$\mathbf{0.00 \pm 0.00}$	98.00 ± 22.37	106.00 ± 11.25	58.50 ± 10.87	$\mathbf{0.00 \pm 0.44}$	$\mathbf{0.00 \pm 0.00}$	2.00 ± 12.34
	Median Extra Edges→ 0	325.50 ± 33.60	309.00 ± 29.77	758.00 ± 65.10	318.50 ± 31.70	817.50 ± 76.77	859.50 ± 86.60	$\mathbf{1.50 \pm 7.05}$
	Median Missing Edges→ 0	$\mathbf{0.00 \pm 0.00}$	$\mathbf{0.00 \pm 0.00}$	$\mathbf{0.00 \pm 0.00}$	$\mathbf{0.00 \pm 0.00}$	$\mathbf{0.00 \pm 0.00}$	$\mathbf{0.00 \pm 0.00}$	0.00 ± 15.72

Table 15: Comprehensive evaluation results—including edge, node, and structural metrics—for all benchmark models on **Aion**. $\rightarrow 0$ indicates that values closer to 0 are better. $\uparrow$ denotes that higher values are better, while $\downarrow$ denotes that lower values are better. **Bold** indicates best, the second-best is underlined.

	Metric	HTGN	ROLAND	VGRNN	TGCN	GCLSTM	EvolveGCN	TopoGED
Edge Evaluation	oo-bank Precision↑	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	$\mathbf{0.13 \pm 0.35}$	0.00 ± 0.00	0.07 ± 0.03
	oo-bank Recall↑	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.01	0.00 ± 0.00	$\mathbf{0.16 \pm 0.08}$
	oo-bank F1↑	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.01 ± 0.01	0.00 ± 0.00	$\mathbf{0.09 \pm 0.04}$
	oo-nobank Precision↑	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.02 ± 0.09	0.00 ± 0.00	$\mathbf{0.03 \pm 0.03}$
	oo-nobank Recall↑	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	$\mathbf{0.01 \pm 0.02}$
	oo-nobank F1↑	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	$\mathbf{0.02 \pm 0.02}$
	Edge Precision↑	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	$\mathbf{0.15 \pm 0.05}$
	Edge Recall↑	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	$\mathbf{0.15 \pm 0.06}$
	Edge F1↑	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	$\mathbf{0.15 \pm 0.05}$
	Num o-n Predicted→ 0	-0.74 ± 0.07	-0.79 ± 0.07	0.52 ± 0.77	-0.84 ± 0.11	-0.17 ± 0.22	-0.47 ± 0.36	$\mathbf{0.02 \pm 0.22}$
	Num n-n Predicted→ 0	202.57 ± 177.05	206.59 ± 200.54	174.04 ± 157.08	260.36 ± 206.20	176.97 ± 153.41	196.50 ± 172.56	$\mathbf{1.05 \pm 3.46}$
Node Evaluation	Precision Nodes↑	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.02 ± 0.01	0.01 ± 0.02	$\mathbf{0.45 \pm 0.06}$
	Recall Nodes↑	0.00 ± 0.00	0.01 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.01 ± 0.00	0.00 ± 0.00	$\mathbf{0.45 \pm 0.07}$
	F1 Nodes↑	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.01 ± 0.00	0.00 ± 0.01	$\mathbf{0.45 \pm 0.05}$
	Precision Old Nodes↑	0.01 ± 0.02	0.09 ± 0.03	0.08 ± 0.08	0.00 ± 0.01	$\mathbf{0.35 \pm 0.11}$	0.18 ± 0.21	0.14 ± 0.04
	Recall Old Nodes↑	0.00 ± 0.00	0.01 ± 0.01	0.00 ± 0.00	0.00 ± 0.00	0.01 ± 0.00	0.00 ± 0.01	$\mathbf{0.13 \pm 0.04}$
	F1 Old Nodes↑	0.00 ± 0.00	0.02 ± 0.01	0.01 ± 0.01	0.00 ± 0.00	0.02 ± 0.01	0.01 ± 0.01	$\mathbf{0.13 \pm 0.04}$
	New Nodes Predicted↑	5.77 ± 1.39	4.28 ± 1.34	0.11 ± 0.36	5.03 ± 3.45	-0.33 ± 0.13	-0.50 ± 0.11	$\mathbf{0.01 \pm 0.21}$
Structure Evaluation	Avg Node Degree→ 0	-0.09 ± 0.10	0.17 ± 0.12	5.35 ± 1.64	1.33 ± 2.48	7.78 ± 0.98	11.41 ± 2.50	$\mathbf{0.01 \pm 0.06}$
	Unique Degree Count→ 0	$\mathbf{0.02 \pm 0.20}$	0.08 ± 0.45	0.88 ± 0.61	0.46 ± 1.36	3.11 ± 0.84	2.69 ± 0.81	-0.17 ± 0.21
	Degree Centrality→ 0	-0.66 ± 0.08	-0.42 ± 0.21	14.79 ± 9.11	3.16 ± 8.66	31.15 ± 7.68	63.10 ± 25.23	$\mathbf{0.05 \pm 0.22}$
	Assortativity Coefficient→ 0	-2.60 ± 0.44	1.12 ± 0.53	0.87 ± 0.46	1.27 ± 0.83	0.19 ± 0.38	$\mathbf{-0.05 \pm 0.31}$	1.08 ± 0.37
	Clustering Coefficient→ 0	10.20 ± 11.05	$\mathbf{5.57 \pm 11.05}$	76.65 ± 82.51	12.08 ± 30.30	59.78 ± 63.33	66.35 ± 71.37	7.38 ± 10.13
	Density→ 0	-0.66 ± 0.08	-0.42 ± 0.21	14.79 ± 9.11	3.16 ± 8.66	31.15 ± 7.68	63.10 ± 25.23	$\mathbf{0.05 \pm 0.22}$
	Num Triangles→ 0	87.45 ± 65.09	48.10 ± 66.90	1912.40 ± 1276.79	490.73 ± 767.46	2594.00 ± 1728.90	3245.35 ± 2425.08	$\mathbf{15.53 \pm 16.14}$
	Descriptor Norm→ 0	1330.29 ± 301.01	1164.38 ± 502.01	747.93 ± 226.62	1902.54 ± 1532.02	790.99 ± 218.30	909.13 ± 280.69	$\mathbf{279.83 \pm 115.95}$
	Median Extra Nodes→ 0	475.50 ± 19.30	321.50 ± 22.01	$\mathbf{0.00 \pm 0.00}$	353.50 ± 104.50	$\mathbf{0.00 \pm 0.00}$	$\mathbf{0.00 \pm 0.00}$	0.00 ± 3.01
	Median Missing Nodes→ 0	$\mathbf{0.00 \pm 0.00}$	$\mathbf{0.00 \pm 0.00}$	146.50 ± 12.55	$\mathbf{0.00 \pm 0.00}$	184.50 ± 22.56	200.00 ± 28.53	8.50 ± 6.23
	Median Extra Edges→ 0	379.50 ± 25.05	363.50 ± 26.38	409.00 ± 57.09	465.00 ± 65.47	378.50 ± 24.70	386.00 ± 26.84	$\mathbf{2.50 \pm 5.93}$
	Median Missing Edges→ 0	$\mathbf{0.00 \pm 0.00}$	$\mathbf{0.00 \pm 0.00}$	$\mathbf{0.00 \pm 0.00}$	$\mathbf{0.00 \pm 0.00}$	$\mathbf{0.00 \pm 0.00}$	$\mathbf{0.00 \pm 0.00}$	4.00 ± 6.50

Table 16: Comprehensive evaluation results—including edge, node, and structural metrics—for all benchmark models on **Aragon**. $\rightarrow 0$ indicates that values closer to 0 are better. $\uparrow$ denotes that higher values are better, while $\downarrow$ denotes that lower values are better. **Bold** indicates best, the second-best is underlined. OOM indicate out-of-memory.

	Metric	HTGN	ROLAND	VGRNN	TGCN	GCLSTM	EvolveGCN	TopoGED
Edge Evaluation	oo-bank Precision↑	0.00 ± 0.00	0.00 ± 0.00	OOM	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	$\mathbf{0.07 \pm 0.06}$
	oo-bank Recall↑	0.00 ± 0.00	0.00 ± 0.00	OOM	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	$\mathbf{0.13 \pm 0.14}$
	oo-bank F1↑	0.00 ± 0.00	0.00 ± 0.00	OOM	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	$\mathbf{0.08 \pm 0.08}$
	oo-nobank Precision↑	0.00 ± 0.00	0.00 ± 0.00	OOM	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	$\mathbf{0.00 \pm 0.01}$
	oo-nobank Recall↑	0.00 ± 0.00	0.00 ± 0.00	OOM	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	$\mathbf{0.00 \pm 0.01}$
	oo-nobank F1↑	0.00 ± 0.00	0.00 ± 0.00	OOM	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	$\mathbf{0.00 \pm 0.01}$
	Edge Precision↑	0.00 ± 0.00	0.00 ± 0.00	OOM	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	$\mathbf{0.22 \pm 0.08}$
	Edge Recall↑	0.00 ± 0.00	0.00 ± 0.00	OOM	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	$\mathbf{0.23 \pm 0.09}$
	Edge F1↑	0.00 ± 0.00	0.00 ± 0.00	OOM	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	$\mathbf{0.22 \pm 0.08}$
	Num o-n Predicted→ 0	-0.63 ± 0.10	-0.87 ± 0.04	OOM	-0.90 ± 0.04	-0.78 ± 0.20	-0.77 ± 0.33	$\mathbf{0.07 \pm 0.29}$
	Num n-n Predicted→ 0	157.99 ± 97.18	173.19 ± 108.70	OOM	171.62 ± 104.11	335.32 ± 196.71	168.11 ± 109.39	$\mathbf{0.58 \pm 1.81}$
Node Evaluation	Precision Nodes↑	0.00 ± 0.00	0.00 ± 0.00	OOM	0.00 ± 0.00	0.01 ± 0.00	0.00 ± 0.00	$\mathbf{0.51 \pm 0.06}$
	Recall Nodes↑	0.00 ± 0.00	0.00 ± 0.00	OOM	0.00 ± 0.00	0.01 ± 0.01	0.00 ± 0.01	$\mathbf{0.53 \pm 0.08}$
	F1 Nodes↑	0.00 ± 0.00	0.00 ± 0.00	OOM	0.00 ± 0.00	0.01 ± 0.00	0.00 ± 0.00	$\mathbf{0.51 \pm 0.04}$
	Precision Old Nodes↑	0.00 ± 0.01	0.01 ± 0.03	OOM	0.00 ± 0.00	$\mathbf{0.28 \pm 0.10}$	0.06 ± 0.09	0.16 ± 0.04
	Recall Old Nodes↑	0.00 ± 0.00	0.00 ± 0.00	OOM	0.00 ± 0.00	0.02 ± 0.01	0.01 ± 0.02	$\mathbf{0.16 \pm 0.05}$
	F1 Old Nodes↑	0.00 ± 0.00	0.00 ± 0.01	OOM	0.00 ± 0.00	0.04 ± 0.02	0.01 ± 0.02	$\mathbf{0.16 \pm 0.04}$
	New Nodes Predicted↑	8.14 ± 2.58	3.33 ± 1.38	OOM	0.08 ± 0.44	2.53 ± 1.10	3.36 ± 1.30	$\mathbf{0.06 \pm 0.27}$
Structure Evaluation	Avg Node Degree→ 0	-0.44 ± 0.03	0.26 ± 0.28	OOM	4.21 ± 1.17	2.14 ± 1.14	0.25 ± 0.23	$\mathbf{0.01 \pm 0.06}$
	Unique Degree Count→ 0	-0.79 ± 0.11	-0.50 ± 0.24	OOM	2.53 ± 0.95	0.45 ± 0.70	1.07 ± 0.69	$\mathbf{0.13 \pm 0.24}$
	Degree Centrality→ 0	-0.87 ± 0.04	-0.31 ± 0.47	OOM	10.29 ± 4.69	1.09 ± 1.33	-0.37 ± 0.28	$\mathbf{0.01 \pm 0.27}$
	Assortativity Coefficient→ 0	-2.64 ± 1.47	2.08 ± 0.47	OOM	0.90 ± 0.35	1.84 ± 0.44	$\mathbf{-0.07 \pm 1.11}$	0.79 ± 0.33
	Clustering Coefficient→ 0	$\mathbf{-0.82 \pm 0.38}$	-0.83 ± 0.38	OOM	23.32 ± 19.28	29.93 ± 23.87	4.42 ± 5.72	1.01 ± 2.22
	Density→ 0	-0.87 ± 0.04	-0.31 ± 0.47	OOM	10.29 ± 4.69	1.09 ± 1.33	-0.37 ± 0.28	$\mathbf{0.01 \pm 0.27}$
	Num Triangles→ 0	$\mathbf{-0.75 \pm 0.57}$	-0.83 ± 0.38	OOM	595.75 ± 473.80	463.09 ± 516.32	70.68 ± 74.09	2.62 ± 3.67
	Descriptor Norm→ 0	1676.71 ± 786.88	523.55 ± 295.90	OOM	296.62 ± 130.53	516.47 ± 178.13	419.98 ± 232.31	$\mathbf{127.58 \pm 86.93}$
	Median Extra Nodes→ 0	377.50 ± 24.71	117.50 ± 10.36	OOM	$\mathbf{0.00 \pm 0.00}$	84.50 ± 6.86	120.50 ± 10.29	8.50 ± 4.56
	Median Missing Nodes→ 0	$\mathbf{0.00 \pm 0.00}$	$\mathbf{0.00 \pm 0.00}$	OOM	55.00 ± 6.53	$\mathbf{0.00 \pm 0.00}$	$\mathbf{0.00 \pm 0.00}$	0.00 ± 1.05
	Median Extra Edges→ 0	146.00 ± 12.86	152.50 ± 11.25	OOM	145.50 ± 13.10	377.50 ± 26.38	145.50 ± 13.57	$\mathbf{9.00 \pm 4.31}$
	Median Missing Edges→ 0	$\mathbf{0.00 \pm 0.00}$	$\mathbf{0.00 \pm 0.00}$	OOM	$\mathbf{0.00 \pm 0.00}$	$\mathbf{0.00 \pm 0.00}$	$\mathbf{0.00 \pm 0.00}$	0.00 ± 0.91

Table 17: Comprehensive evaluation results—including edge, node, and structural metrics—for all benchmark models on **Bancor**. $\rightarrow 0$ indicates that values closer to 0 are better. $\uparrow$ denotes that higher values are better, while $\downarrow$ denotes that lower values are better. **Bold** indicates best, the second-best is underlined. OOM indicate out-of-memory.

	Metric	HTGN	ROLAND	VGRNN	TGCN	GCLSTM	EvolveGCN	TopoGED
Edge Evaluation	oo-bank Precision↑	0.00 ± 0.00	0.00 ± 0.00	OOM	0.00 ± 0.00	0.23 ± 0.42	0.01 ± 0.05	$\mathbf{0.25 \pm 0.11}$
	oo-bank Recall↑	0.00 ± 0.00	0.00 ± 0.00	OOM	0.00 ± 0.00	0.01 ± 0.01	0.00 ± 0.00	$\mathbf{0.34 \pm 0.15}$
	oo-bank F1↑	0.00 ± 0.00	0.00 ± 0.00	OOM	0.00 ± 0.00	0.01 ± 0.03	0.00 ± 0.00	$\mathbf{0.27 \pm 0.10}$
	oo-nobank Precision↑	0.00 ± 0.00	0.00 ± 0.00	OOM	0.00 ± 0.00	$\mathbf{0.01 \pm 0.05}$	0.00 ± 0.00	$\mathbf{0.01 \pm 0.03}$
	oo-nobank Recall↑	0.00 ± 0.00	0.00 ± 0.00	OOM	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	$\mathbf{0.00 \pm 0.02}$
	oo-nobank F1↑	0.00 ± 0.00	0.00 ± 0.00	OOM	0.00 ± 0.00	0.00 ± 0.01	0.00 ± 0.00	$\mathbf{0.01 \pm 0.03}$
	Edge Precision↑	0.00 ± 0.00	0.00 ± 0.00	OOM	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	$\mathbf{0.15 \pm 0.08}$
	Edge Recall↑	0.00 ± 0.00	0.00 ± 0.00	OOM	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	$\mathbf{0.14 \pm 0.07}$
	Edge F1↑	0.00 ± 0.00	0.00 ± 0.00	OOM	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	$\mathbf{0.14 \pm 0.07}$
	Num o-n Predicted→ 0	-0.45 ± 0.22	-0.29 ± 0.45	OOM	0.34 ± 0.53	0.59 ± 0.51	-0.11 ± 0.40	$\mathbf{0.04 \pm 0.35}$
	Num n-n Predicted→ 0	69.62 ± 65.14	64.20 ± 59.69	OOM	55.66 ± 52.53	107.34 ± 97.87	64.87 ± 61.32	$\mathbf{0.56 \pm 1.80}$
Node Evaluation	Precision Nodes↑	0.00 ± 0.00	0.01 ± 0.01	OOM	0.00 ± 0.00	0.03 ± 0.01	0.03 ± 0.02	$\mathbf{0.58 \pm 0.09}$
	Recall Nodes↑	0.00 ± 0.00	0.01 ± 0.00	OOM	0.00 ± 0.00	0.02 ± 0.01	0.01 ± 0.01	$\mathbf{0.57 \pm 0.10}$
	F1 Nodes↑	0.00 ± 0.00	0.01 ± 0.00	OOM	0.00 ± 0.00	0.03 ± 0.01	0.02 ± 0.01	$\mathbf{0.57 \pm 0.07}$
	Precision Old Nodes↑	0.00 ± 0.00	$\mathbf{0.62 \pm 0.44}$	OOM	0.01 ± 0.03	0.51 ± 0.10	0.34 ± 0.16	0.33 ± 0.09
	Recall Old Nodes↑	0.00 ± 0.00	0.01 ± 0.01	OOM	0.00 ± 0.00	0.04 ± 0.01	0.03 ± 0.01	$\mathbf{0.32 \pm 0.08}$
	F1 Old Nodes↑	0.00 ± 0.00	0.02 ± 0.01	OOM	0.00 ± 0.01	0.07 ± 0.02	0.05 ± 0.03	$\mathbf{0.32 \pm 0.07}$
	New Nodes Predicted↑	8.58 ± 3.85	0.48 ± 0.61	OOM	$\mathbf{-0.01 \pm 0.33}$	0.41 ± 0.50	0.27 ± 0.47	0.03 ± 0.31
Structure Evaluation	Avg Node Degree→ 0	-0.39 ± 0.05	3.18 ± 1.22	OOM	4.44 ± 0.87	6.20 ± 1.73	3.69 ± 1.31	$\mathbf{0.01 \pm 0.09}$
	Unique Degree Count→ 0	-0.63 ± 0.07	1.31 ± 0.59	OOM	1.83 ± 0.66	1.72 ± 0.92	1.63 ± 0.65	$\mathbf{0.16 \pm 0.24}$
	Degree Centrality→ 0	-0.84 ± 0.07	7.45 ± 6.13	OOM	11.67 ± 6.18	12.28 ± 9.08	8.53 ± 6.83	$\mathbf{0.08 \pm 0.46}$
	Assortativity Coefficient→ 0	-2.25 ± 0.42	1.39 ± 0.57	OOM	$\mathbf{0.94 \pm 0.42}$	1.81 ± 0.70	-2.27 ± 0.74	1.13 ± 0.52
	Clustering Coefficient→ 0	$\mathbf{-0.84 \pm 0.34}$	34.97 ± 42.92	OOM	40.36 ± 48.15	44.31 ± 50.13	29.82 ± 35.45	2.63 ± 4.87
	Density→ 0	-0.84 ± 0.07	7.45 ± 6.13	OOM	11.67 ± 6.18	12.28 ± 9.08	8.53 ± 6.83	$\mathbf{0.08 \pm 0.46}$
	Num Triangles→ 0	$\mathbf{-0.74 \pm 0.58}$	374.60 ± 295.06	OOM	492.84 ± 468.77	1027.56 ± 1080.19	559.73 ± 549.70	5.15 ± 6.79
	Descriptor Norm→ 0	1656.96 ± 1283.31	319.63 ± 255.20	OOM	365.52 ± 308.79	500.71 ± 299.54	374.58 ± 338.75	$\mathbf{171.73 \pm 230.58}$
	Median Extra Nodes→ 0	380.00 ± 20.37	$\mathbf{0.00 \pm 0.00}$	OOM	$\mathbf{0.00 \pm 0.00}$	$\mathbf{0.00 \pm 0.00}$	$\mathbf{0.00 \pm 0.00}$	0.00 ± 2.21
	Median Missing Nodes→ 0	$\mathbf{0.00 \pm 0.00}$	47.50 ± 5.96	OOM	68.00 ± 4.53	45.50 ± 4.88	56.00 ± 4.12	1.50 ± 1.74
	Median Extra Edges→ 0	154.00 ± 9.87	158.00 ± 13.16	OOM	153.00 ± 10.12	356.50 ± 11.58	160.00 ± 9.92	$\mathbf{1.50 \pm 2.91}$
	Median Missing Edges→ 0	$\mathbf{0.00 \pm 0.00}$	$\mathbf{0.00 \pm 0.00}$	OOM	$\mathbf{0.00 \pm 0.00}$	$\mathbf{0.00 \pm 0.00}$	$\mathbf{0.00 \pm 0.00}$	0.00 ± 1.76

Table 18: Comprehensive evaluation results—including edge, node, and structural metrics—for all benchmark models on **Centra**. $\rightarrow 0$ indicates that values closer to 0 are better. $\uparrow$ denotes that higher values are better, while $\downarrow$ denotes that lower values are better. **Bold** indicates best, the second-best is underlined.

	Metric	HTGN	ROLAND	VGRNN	TGCN	GCLSTM	EvolveGCN	TopoGED
Edge Evaluation	oo-bank Precision↑	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	$\mathbf{0.12 \pm 0.22}$	0.05 ± 0.22	0.09 ± 0.09
	oo-bank Recall↑	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.01 ± 0.02	0.00 ± 0.00	$\mathbf{0.18 \pm 0.21}$
	oo-bank F1↑	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.02 ± 0.04	0.00 ± 0.01	$\mathbf{0.11 \pm 0.11}$
	oo-nobank Precision↑	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	$\mathbf{0.01 \pm 0.02}$
	oo-nobank Recall↑	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	$\mathbf{0.01 \pm 0.02}$
	oo-nobank F1↑	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	$\mathbf{0.01 \pm 0.01}$
	Edge Precision↑	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	$\mathbf{0.07 \pm 0.06}$
	Edge Recall↑	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	$\mathbf{0.08 \pm 0.07}$
	Edge F1↑	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	$\mathbf{0.07 \pm 0.06}$
	Num o-n Predicted→ 0	-0.19 ± 0.64	$\mathbf{0.05 \pm 1.48}$	0.64 ± 2.27	-0.87 ± 0.18	0.52 ± 1.26	-0.26 ± 0.66	0.46 ± 0.92
	Num n-n Predicted→ 0	419.16 ± 554.55	500.92 ± 743.93	544.65 ± 652.79	153.58 ± 101.27	473.27 ± 499.76	423.32 ± 569.51	$\mathbf{0.74 \pm 1.74}$
Node Evaluation	Precision Nodes↑	0.00 ± 0.00	0.00 ± 0.00	0.01 ± 0.01	0.00 ± 0.00	0.02 ± 0.01	0.01 ± 0.01	$\mathbf{0.47 \pm 0.15}$
	Recall Nodes↑	0.00 ± 0.00	0.01 ± 0.01	0.01 ± 0.01	0.00 ± 0.00	0.01 ± 0.01	0.01 ± 0.01	$\mathbf{0.56 \pm 0.22}$
	F1 Nodes↑	0.00 ± 0.00	0.01 ± 0.00	0.01 ± 0.01	0.00 ± 0.00	0.01 ± 0.01	0.01 ± 0.01	$\mathbf{0.47 \pm 0.14}$
	Precision Old Nodes↑	0.00 ± 0.00	0.05 ± 0.04	$\mathbf{0.23 \pm 0.33}$	0.00 ± 0.00	0.17 ± 0.08	0.22 ± 0.29	$\mathbf{0.23 \pm 0.13}$
	Recall Old Nodes↑	0.00 ± 0.01	0.02 ± 0.02	0.01 ± 0.01	0.00 ± 0.00	0.03 ± 0.02	0.02 ± 0.02	$\mathbf{0.26 \pm 0.13}$
	F1 Old Nodes↑	0.00 ± 0.00	0.02 ± 0.02	0.02 ± 0.02	0.00 ± 0.00	0.05 ± 0.03	0.03 ± 0.04	$\mathbf{0.21 \pm 0.11}$
	New Nodes Predicted↑	10.35 ± 6.75	5.31 ± 6.29	1.75 ± 5.46	2.89 ± 2.64	1.31 ± 2.09	0.96 ± 3.44	$\mathbf{0.35 \pm 0.76}$
Structure Evaluation	Avg Node Degree→ 0	-0.39 ± 0.17	0.30 ± 0.55	6.05 ± 5.77	1.49 ± 1.64	4.89 ± 2.68	4.58 ± 3.51	$\mathbf{0.03 \pm 0.12}$
	Unique Degree Count→ 0	-0.46 ± 0.48	0.43 ± 1.40	0.53 ± 0.74	$\mathbf{-0.05 \pm 0.73}$	2.15 ± 1.72	2.14 ± 1.23	0.06 ± 0.30
	Degree Centrality→ 0	-0.79 ± 0.30	$\mathbf{0.52 \pm 5.05}$	21.35 ± 45.22	3.04 ± 4.25	20.43 ± 39.18	24.73 ± 56.03	0.56 ± 3.55
	Assortativity Coefficient→ 0	-1.58 ± 0.61	0.25 ± 2.47	0.26 ± 1.95	0.46 ± 3.29	0.07 ± 2.20	-0.94 ± 0.63	$\mathbf{-0.04 \pm 1.53}$
	Clustering Coefficient→ 0	4.89 ± 18.04	14.82 ± 63.29	56.31 ± 135.01	6.74 ± 23.95	46.57 ± 103.10	35.26 ± 83.73	$\mathbf{2.13 \pm 4.24}$
	Density→ 0	-0.79 ± 0.30	$\mathbf{0.52 \pm 5.05}$	21.35 ± 45.22	3.04 ± 4.25	20.43 ± 39.18	24.73 ± 56.03	0.56 ± 3.55
	Num Triangles→ 0	138.66 ± 722.69	486.66 ± 1408.43	1843.29 ± 4451.19	126.88 ± 152.34	2946.97 ± 8072.72	1808.73 ± 3466.10	$\mathbf{16.99 \pm 43.03}$
	Descriptor Norm→ 0	3578.26 ± 4205.39	1862.97 ± 2565.60	2326.09 ± 3778.83	$\mathbf{650.07 \pm 1494.33}$	1758.78 ± 2467.73	1995.73 ± 3215.58	1042.17 ± 1701.40
	Median Extra Nodes→ 0	411.00 ± 73.46	168.00 ± 34.40	$\mathbf{0.00 \pm 1.30}$	0.00 ± 31.00	0.00 ± 1.77	$\mathbf{0.00 \pm 0.00}$	19.50 ± 11.98
	Median Missing Nodes→ 0	$\mathbf{0.00 \pm 0.00}$	$\mathbf{0.00 \pm 0.00}$	49.50 ± 29.18	1.00 ± 7.40	21.50 ± 23.48	58.00 ± 10.03	0.00 ± 1.68
	Median Extra Edges→ 0	154.50 ± 40.46	186.50 ± 53.45	361.50 ± 66.36	115.00 ± 18.19	366.50 ± 49.04	151.50 ± 41.55	$\mathbf{26.50 \pm 19.25}$
	Median Missing Edges→ 0	$\mathbf{0.00 \pm 0.00}$	$\mathbf{0.00 \pm 0.00}$	$\mathbf{0.00 \pm 0.00}$	0.00 ± 2.12	$\mathbf{0.00 \pm 0.00}$	$\mathbf{0.00 \pm 0.00}$	0.00 ± 1.06

Table 19: Comprehensive evaluation results—including edge, node, and structural metrics—for all benchmark models on **Cindicator**. $\rightarrow 0$ indicates that values closer to 0 are better. $\uparrow$ denotes that higher values are better, while $\downarrow$ denotes that lower values are better. **Bold** indicates best, the second-best is underlined.

	Metric	HTGN	ROLAND	VGRNN	TGCN	GCLSTM	EvolveGCN	TopoGED
Edge Evaluation	oo-bank Precision↑	0.00 ± 0.00	$\mathbf{0.15 \pm 0.36}$	0.06 ± 0.24	0.00 ± 0.00	$\mathbf{0.15 \pm 0.36}$	0.00 ± 0.00	0.11 ± 0.07
	oo-bank Recall↑	0.00 ± 0.00	0.00 ± 0.01	0.00 ± 0.01	0.00 ± 0.00	0.00 ± 0.01	0.00 ± 0.00	$\mathbf{0.18 \pm 0.11}$
	oo-bank F1↑	0.00 ± 0.00	0.01 ± 0.02	0.00 ± 0.01	0.00 ± 0.00	0.01 ± 0.02	0.00 ± 0.00	$\mathbf{0.13 \pm 0.07}$
	oo-nobank Precision↑	0.00 ± 0.00	0.00 ± 0.01	0.00 ± 0.00	0.00 ± 0.00	0.01 ± 0.06	0.00 ± 0.00	$\mathbf{0.02 \pm 0.03}$
	oo-nobank Recall↑	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	$\mathbf{0.01 \pm 0.02}$
	oo-nobank F1↑	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.01	0.00 ± 0.00	$\mathbf{0.01 \pm 0.02}$
	Edge Precision↑	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	$\mathbf{0.12 \pm 0.07}$
	Edge Recall↑	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	$\mathbf{0.13 \pm 0.07}$
	Edge F1↑	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	$\mathbf{0.13 \pm 0.07}$
	Num o-n Predicted→ 0	-0.70 ± 0.08	3.67 ± 1.44	-0.51 ± 0.42	-1.00 ± 0.00	0.22 ± 0.25	-0.70 ± 0.17	$\mathbf{0.05 \pm 0.26}$
	Num n-n Predicted→ 0	278.15 ± 150.62	11.05 ± 11.11	268.33 ± 146.33	293.46 ± 159.21	228.10 ± 122.24	276.82 ± 154.10	$\mathbf{0.52 \pm 1.58}$
Node Evaluation	Precision Nodes↑	0.00 ± 0.00	0.00 ± 0.00	0.01 ± 0.01	0.00 ± 0.00	0.03 ± 0.01	0.01 ± 0.00	$\mathbf{0.51 \pm 0.06}$
	Recall Nodes↑	0.00 ± 0.00	0.01 ± 0.00	0.01 ± 0.01	0.00 ± 0.00	0.01 ± 0.00	0.01 ± 0.01	$\mathbf{0.52 \pm 0.06}$
	F1 Nodes↑	0.00 ± 0.00	0.01 ± 0.00	0.01 ± 0.01	0.00 ± 0.00	0.02 ± 0.01	0.01 ± 0.00	$\mathbf{0.51 \pm 0.04}$
	Precision Old Nodes↑	0.00 ± 0.00	0.17 ± 0.10	0.39 ± 0.33	0.00 ± 0.00	$\mathbf{0.55 \pm 0.16}$	0.20 ± 0.09	0.20 ± 0.05
	Recall Old Nodes↑	0.00 ± 0.00	0.02 ± 0.01	0.01 ± 0.01	0.00 ± 0.00	0.02 ± 0.01	0.02 ± 0.01	$\mathbf{0.20 \pm 0.04}$
	F1 Old Nodes↑	0.00 ± 0.00	0.03 ± 0.01	0.02 ± 0.02	0.00 ± 0.00	0.04 ± 0.01	0.04 ± 0.02	$\mathbf{0.20 \pm 0.04}$
	New Nodes Predicted↑	8.54 ± 1.90	4.60 ± 1.63	$\mathbf{0.00 \pm 0.41}$	0.22 ± 0.22	-0.21 ± 0.14	3.82 ± 1.35	0.04 ± 0.23
Structure Evaluation	Avg Node Degree→ 0	-0.39 ± 0.03	0.05 ± 0.07	5.81 ± 3.09	3.91 ± 0.55	6.43 ± 0.52	0.24 ± 0.27	$\mathbf{0.00 \pm 0.06}$
	Unique Degree Count→ 0	-0.45 ± 0.11	$\mathbf{-0.06 \pm 0.24}$	0.45 ± 0.56	2.33 ± 0.70	2.30 ± 0.66	1.18 ± 0.55	0.17 ± 0.21
	Degree Centrality→ 0	-0.85 ± 0.03	-0.54 ± 0.14	21.57 ± 22.64	8.64 ± 1.77	20.63 ± 3.97	-0.35 ± 0.35	$\mathbf{-0.00 \pm 0.19}$
	Assortativity Coefficient→ 0	-1.95 ± 0.26	1.03 ± 0.31	0.33 ± 0.49	0.45 ± 0.23	0.53 ± 0.24	$\mathbf{-0.13 \pm 0.35}$	0.32 ± 0.23
	Clustering Coefficient→ 0	6.44 ± 22.00	$\mathbf{2.41 \pm 9.06}$	68.92 ± 217.82	64.29 ± 184.65	66.34 ± 198.51	5.82 ± 25.51	3.44 ± 11.07
	Density→ 0	-0.85 ± 0.03	-0.54 ± 0.14	21.57 ± 22.64	8.64 ± 1.77	20.63 ± 3.97	-0.35 ± 0.35	$\mathbf{-0.00 \pm 0.19}$
	Num Triangles→ 0	16.02 ± 15.07	18.88 ± 17.75	1287.65 ± 1062.84	911.88 ± 726.83	1130.89 ± 770.31	74.73 ± 242.41	$\mathbf{7.44 \pm 8.97}$
	Descriptor Norm→ 0	1599.32 ± 396.49	848.61 ± 387.64	503.62 ± 232.09	379.57 ± 100.96	437.60 ± 89.09	555.81 ± 203.93	$\mathbf{133.74 \pm 56.00}$
	Median Extra Nodes→ 0	490.50 ± 18.93	213.50 ± 22.82	$\mathbf{0.00 \pm 0.00}$	$\mathbf{0.00 \pm 0.00}$	$\mathbf{0.00 \pm 0.00}$	156.00 ± 8.67	8.00 ± 6.02
	Median Missing Nodes→ 0	$\mathbf{0.00 \pm 0.00}$	$\mathbf{0.00 \pm 0.00}$	93.00 ± 10.35	74.00 ± 5.71	103.00 ± 6.58	$\mathbf{0.00 \pm 0.00}$	0.00 ± 3.62
	Median Extra Edges→ 0	226.50 ± 17.21	238.50 ± 20.94	236.50 ± 17.67	223.00 ± 17.48	234.50 ± 15.77	223.00 ± 18.10	$\mathbf{4.00 \pm 6.47}$
	Median Missing Edges→ 0	$\mathbf{0.00 \pm 0.00}$	$\mathbf{0.00 \pm 0.00}$	$\mathbf{0.00 \pm 0.00}$	$\mathbf{0.00 \pm 0.00}$	$\mathbf{0.00 \pm 0.00}$	$\mathbf{0.00 \pm 0.00}$	0.00 ± 5.21

Table 20: Comprehensive evaluation results—including edge, node, and structural metrics—for all benchmark models on **Coindash**. $\rightarrow 0$ indicates that values closer to 0 are better. $\uparrow$ denotes that higher values are better, while $\downarrow$ denotes that lower values are better. **Bold** indicates best, the second-best is underlined.

	Metric	HTGN	ROLAND	VGRNN	TGCN	GCLSTM	EvolveGCN	TopoGED
Edge Evaluation	oo-bank Precision↑	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.05 ± 0.15	0.00 ± 0.00	$\mathbf{0.20 \pm 0.10}$
	oo-bank Recall↑	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.01	0.00 ± 0.00	$\mathbf{0.28 \pm 0.14}$
	oo-bank F1↑	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.01 ± 0.02	0.00 ± 0.00	$\mathbf{0.22 \pm 0.10}$
	oo-nobank Precision↑	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	$\mathbf{0.02 \pm 0.05}$
	oo-nobank Recall↑	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	$\mathbf{0.01 \pm 0.04}$
	oo-nobank F1↑	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	$\mathbf{0.02 \pm 0.04}$
	Edge Precision↑	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	$\mathbf{0.13 \pm 0.06}$
	Edge Recall↑	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	$\mathbf{0.13 \pm 0.07}$
	Edge F1↑	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	$\mathbf{0.13 \pm 0.06}$
	Num o-n Predicted→ 0	0.19 ± 0.78	-0.41 ± 0.32	-0.34 ± 1.58	-0.93 ± 0.40	0.58 ± 2.12	-0.28 ± 0.69	$\mathbf{0.13 \pm 0.71}$
	Num n-n Predicted→ 0	179.27 ± 383.82	187.30 ± 95.05	240.66 ± 413.80	203.72 ± 402.28	299.16 ± 368.46	212.74 ± 530.69	$\mathbf{0.26 \pm 1.29}$
Node Evaluation	Precision Nodes↑	0.00 ± 0.00	0.01 ± 0.01	0.01 ± 0.02	0.00 ± 0.00	0.03 ± 0.01	0.02 ± 0.02	$\mathbf{0.49 \pm 0.13}$
	Recall Nodes↑	0.00 ± 0.00	0.02 ± 0.02	0.01 ± 0.01	0.00 ± 0.00	0.04 ± 0.02	0.01 ± 0.01	$\mathbf{0.49 \pm 0.14}$
	F1 Nodes↑	0.00 ± 0.00	0.01 ± 0.01	0.01 ± 0.01	0.00 ± 0.00	0.03 ± 0.01	0.01 ± 0.01	$\mathbf{0.47 \pm 0.12}$
	Precision Old Nodes↑	0.00 ± 0.00	0.13 ± 0.09	0.28 ± 0.43	0.00 ± 0.00	$\mathbf{0.77 \pm 0.31}$	0.35 ± 0.40	0.35 ± 0.12
	Recall Old Nodes↑	0.00 ± 0.00	0.03 ± 0.03	0.01 ± 0.01	0.00 ± 0.00	0.05 ± 0.03	0.02 ± 0.02	$\mathbf{0.34 \pm 0.12}$
	F1 Old Nodes↑	0.00 ± 0.00	0.05 ± 0.04	0.02 ± 0.03	0.00 ± 0.00	0.10 ± 0.05	0.03 ± 0.03	$\mathbf{0.33 \pm 0.11}$
	New Nodes Predicted↑	17.20 ± 13.86	10.08 ± 6.57	1.25 ± 2.07	1.49 ± 1.20	4.21 ± 2.90	3.06 ± 12.39	$\mathbf{0.09 \pm 0.63}$
Structure Evaluation	Avg Node Degree→ 0	-0.47 ± 0.06	0.18 ± 0.17	5.83 ± 2.78	3.19 ± 1.83	3.25 ± 3.14	3.32 ± 1.33	$\mathbf{0.01 \pm 0.10}$
	Unique Degree Count→ 0	-0.75 ± 0.28	$\mathbf{-0.14 \pm 0.35}$	0.36 ± 0.46	1.64 ± 0.82	0.39 ± 0.91	1.49 ± 0.76	0.16 ± 0.29
	Degree Centrality→ 0	-0.87 ± 0.09	-0.44 ± 0.76	18.00 ± 24.20	8.09 ± 16.17	5.51 ± 17.65	9.74 ± 17.48	$\mathbf{0.21 \pm 1.07}$
	Assortativity Coefficient→ 0	-1.95 ± 0.94	1.36 ± 0.83	0.72 ± 0.87	$\mathbf{0.14 \pm 0.52}$	1.47 ± 0.91	0.28 ± 0.50	0.47 ± 0.57
	Clustering Coefficient→ 0	$\mathbf{0.29 \pm 1.68}$	3.31 ± 12.00	29.96 ± 117.24	25.97 ± 101.14	29.59 ± 114.28	22.53 ± 95.11	3.39 ± 14.63
	Density→ 0	-0.87 ± 0.09	-0.44 ± 0.76	18.00 ± 24.20	8.09 ± 16.17	5.51 ± 17.65	9.74 ± 17.48	$\mathbf{0.21 \pm 1.07}$
	Num Triangles→ 0	$\mathbf{4.49 \pm 25.54}$	26.28 ± 25.71	830.21 ± 976.67	486.47 ± 1005.20	703.41 ± 1325.11	368.13 ± 281.53	7.10 ± 8.94
	Descriptor Norm→ 0	1216.42 ± 1642.06	499.46 ± 426.32	405.05 ± 624.87	308.18 ± 658.46	426.25 ± 632.52	488.05 ± 1625.19	$\mathbf{186.88 \pm 532.55}$
	Median Extra Nodes→ 0	235.50 ± 26.04	114.50 ± 8.74	$\mathbf{0.00 \pm 0.00}$	$\mathbf{0.00 \pm 0.00}$	30.00 ± 4.15	$\mathbf{0.00 \pm 0.00}$	0.50 ± 1.95
	Median Missing Nodes→ 0	$\mathbf{0.00 \pm 0.00}$	$\mathbf{0.00 \pm 0.00}$	25.50 ± 5.60	21.00 ± 3.98	$\mathbf{0.00 \pm 0.00}$	24.50 ± 4.13	0.00 ± 1.79
	Median Extra Edges→ 0	86.00 ± 10.55	140.00 ± 10.59	151.00 ± 21.81	85.00 ± 9.91	227.00 ± 24.04	85.50 ± 9.77	$\mathbf{1.50 \pm 2.61}$
	Median Missing Edges→ 0	$\mathbf{0.00 \pm 0.00}$	$\mathbf{0.00 \pm 0.00}$	$\mathbf{0.00 \pm 0.00}$	$\mathbf{0.00 \pm 0.00}$	$\mathbf{0.00 \pm 0.00}$	$\mathbf{0.00 \pm 0.00}$	0.00 ± 1.15

Table 21: Comprehensive evaluation results—including edge, node, and structural metrics—for all benchmark models on **DGD**. $\rightarrow 0$ indicates that values closer to 0 are better. $\uparrow$ denotes that higher values are better, while $\downarrow$ denotes that lower values are better. **Bold** indicates best, the second-best is underlined. OOM indicate out-of-memory.

	Metric	HTGN	ROLAND	VGRNN	TGCN	GCLSTM	EvolveGCN	TopoGED
Edge Evaluation	oo-bank Precision↑	0.00 ± 0.00	0.01 ± 0.10	OOM	0.00 ± 0.00	0.00 ± 0.03	0.00 ± 0.00	$\mathbf{0.14 \pm 0.09}$
	oo-bank Recall↑	0.00 ± 0.00	0.00 ± 0.00	OOM	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	$\mathbf{0.17 \pm 0.11}$
	oo-bank F1↑	0.00 ± 0.00	0.01 ± 0.01	OOM	0.00 ± 0.00	0.01 ± 0.01	0.00 ± 0.00	$\mathbf{0.15 \pm 0.08}$
	oo-nobank Precision↑	0.00 ± 0.00	0.00 ± 0.00	OOM	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	$\mathbf{0.01 \pm 0.03}$
	oo-nobank Recall↑	0.00 ± 0.00	0.00 ± 0.00	OOM	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	$\mathbf{0.01 \pm 0.02}$
	oo-nobank F1↑	0.00 ± 0.00	0.00 ± 0.00	OOM	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	$\mathbf{0.01 \pm 0.02}$
	Edge Precision↑	0.00 ± 0.00	0.00 ± 0.00	OOM	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	$\mathbf{0.08 \pm 0.05}$
	Edge Recall↑	0.00 ± 0.00	0.00 ± 0.00	OOM	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	$\mathbf{0.08 \pm 0.05}$
	Edge F1↑	0.00 ± 0.00	0.00 ± 0.00	OOM	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	$\mathbf{0.08 \pm 0.05}$
	Num o-n Predicted→ 0	-0.71 ± 0.13	-0.65 ± 0.43	OOM	-0.61 ± 0.37	$\mathbf{0.01 \pm 1.20}$	-0.63 ± 0.23	0.17 ± 0.49
	Num n-n Predicted→ 0	186.38 ± 164.41	188.90 ± 166.41	OOM	184.65 ± 168.51	217.60 ± 170.96	183.59 ± 164.55	$\mathbf{0.49 \pm 1.55}$
Node Evaluation	Precision Nodes↑	0.00 ± 0.00	0.02 ± 0.02	OOM	0.00 ± 0.00	0.02 ± 0.01	0.00 ± 0.00	$\mathbf{0.53 \pm 0.11}$
	Recall Nodes↑	0.00 ± 0.00	0.01 ± 0.01	OOM	0.00 ± 0.00	0.02 ± 0.01	0.01 ± 0.01	$\mathbf{0.56 \pm 0.13}$
	F1 Nodes↑	0.00 ± 0.00	0.01 ± 0.01	OOM	0.00 ± 0.00	0.02 ± 0.01	0.01 ± 0.01	$\mathbf{0.53 \pm 0.09}$
	Precision Old Nodes↑	0.02 ± 0.05	$\mathbf{0.42 \pm 0.38}$	OOM	0.00 ± 0.00	0.37 ± 0.19	0.15 ± 0.13	0.33 ± 0.09
	Recall Old Nodes↑	0.00 ± 0.01	0.01 ± 0.01	OOM	0.00 ± 0.00	0.04 ± 0.02	0.02 ± 0.02	$\mathbf{0.34 \pm 0.09}$
	F1 Old Nodes↑	0.00 ± 0.01	0.03 ± 0.03	OOM	0.00 ± 0.00	0.07 ± 0.04	0.04 ± 0.03	$\mathbf{0.32 \pm 0.07}$
	New Nodes Predicted↑	8.39 ± 3.56	0.33 ± 0.59	OOM	0.42 ± 0.54	1.67 ± 1.20	6.73 ± 3.08	$\mathbf{0.14 \pm 0.44}$
Structure Evaluation	Avg Node Degree→ 0	-0.27 ± 0.09	4.46 ± 1.86	OOM	3.89 ± 1.18	2.38 ± 0.94	-0.12 ± 0.11	$\mathbf{0.02 \pm 0.10}$
	Unique Degree Count→ 0	-0.23 ± 0.19	1.57 ± 0.93	OOM	1.79 ± 0.75	1.47 ± 0.67	$\mathbf{-0.02 \pm 0.35}$	0.05 ± 0.32
	Degree Centrality→ 0	-0.78 ± 0.09	11.56 ± 7.42	OOM	9.02 ± 5.27	2.77 ± 2.39	-0.67 ± 0.15	$\mathbf{0.06 \pm 0.56}$
	Assortativity Coefficient→ 0	-2.69 ± 0.89	1.48 ± 1.62	OOM	1.42 ± 1.50	1.39 ± 1.95	-2.46 ± 1.33	$\mathbf{1.04 \pm 0.99}$
	Clustering Coefficient→ 0	5.10 ± 16.92	48.00 ± 88.92	OOM	51.54 ± 123.07	48.06 ± 121.80	12.61 ± 34.74	$\mathbf{3.06 \pm 7.58}$
	Density→ 0	-0.78 ± 0.09	11.56 ± 7.42	OOM	9.02 ± 5.27	2.77 ± 2.39	-0.67 ± 0.15	$\mathbf{0.06 \pm 0.56}$
	Num Triangles→ 0	28.25 ± 79.49	989.03 ± 1802.47	OOM	953.29 ± 1918.38	797.07 ± 1384.43	59.66 ± 82.10	$\mathbf{7.17 \pm 9.37}$
	Descriptor Norm→ 0	1033.25 ± 583.15	364.97 ± 280.28	OOM	350.06 ± 283.24	360.46 ± 264.26	761.41 ± 597.24	$\mathbf{180.59 \pm 205.62}$
	Median Extra Nodes→ 0	297.00 ± 19.20	$\mathbf{0.00 \pm 0.00}$	OOM	$\mathbf{0.00 \pm 0.00}$	7.00 ± 3.74	222.00 ± 18.75	12.00 ± 3.60
	Median Missing Nodes→ 0	$\mathbf{0.00 \pm 0.00}$	52.00 ± 3.64	OOM	50.00 ± 5.57	0.00 ± 1.00	$\mathbf{0.00 \pm 0.00}$	0.00 ± 0.06
	Median Extra Edges→ 0	157.00 ± 13.94	148.00 ± 15.52	OOM	154.00 ± 13.97	232.00 ± 14.32	155.00 ± 14.01	$\mathbf{13.00 \pm 3.87}$
	Median Missing Edges→ 0	$\mathbf{0.00 \pm 0.00}$	$\mathbf{0.00 \pm 0.00}$	OOM	$\mathbf{0.00 \pm 0.00}$	$\mathbf{0.00 \pm 0.00}$	$\mathbf{0.00 \pm 0.00}$	0.00 ± 0.13

Table 22: Comprehensive evaluation results—including edge, node, and structural metrics—for all benchmark models on **Iconomi**. $\rightarrow 0$ indicates that values closer to 0 are better. $\uparrow$ denotes that higher values are better, while $\downarrow$ denotes that lower values are better. **Bold** indicates best, the second-best is underlined. OOM indicate out-of-memory.

	Metric	HTGN	ROLAND	VGRNN	TGCN	GCLSTM	EvolveGCN	TopoGED
Edge Evaluation	oo-bank Precision↑	0.00 ± 0.00	0.00 ± 0.00	OOM	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	$\mathbf{0.19 \pm 0.10}$
	oo-bank Recall↑	0.00 ± 0.00	0.00 ± 0.00	OOM	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	$\mathbf{0.28 \pm 0.15}$
	oo-bank F1↑	0.00 ± 0.00	0.00 ± 0.00	OOM	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	$\mathbf{0.22 \pm 0.09}$
	oo-nobank Precision↑	0.00 ± 0.00	0.00 ± 0.00	OOM	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	$\mathbf{0.01 \pm 0.03}$
	oo-nobank Recall↑	0.00 ± 0.00	0.00 ± 0.00	OOM	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	$\mathbf{0.01 \pm 0.02}$
	oo-nobank F1↑	0.00 ± 0.00	0.00 ± 0.00	OOM	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	$\mathbf{0.01 \pm 0.02}$
	Edge Precision↑	0.00 ± 0.00	0.00 ± 0.00	OOM	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	$\mathbf{0.10 \pm 0.06}$
	Edge Recall↑	0.00 ± 0.00	0.00 ± 0.00	OOM	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	$\mathbf{0.11 \pm 0.06}$
	Edge F1↑	0.00 ± 0.00	0.00 ± 0.00	OOM	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	$\mathbf{0.10 \pm 0.06}$
	Num o-n Predicted→ 0	-0.58 ± 0.41	-0.79 ± 0.32	OOM	$\mathbf{0.02 \pm 0.63}$	-0.53 ± 0.54	-0.75 ± 0.36	0.18 ± 0.54
	Num n-n Predicted→ 0	157.28 ± 145.81	160.25 ± 149.02	OOM	133.04 ± 134.74	303.04 ± 233.97	175.42 ± 285.77	$\mathbf{0.38 \pm 1.37}$
Node Evaluation	Precision Nodes↑	0.00 ± 0.00	0.01 ± 0.01	OOM	0.00 ± 0.00	0.01 ± 0.00	0.00 ± 0.00	$\mathbf{0.52 \pm 0.11}$
	Recall Nodes↑	0.00 ± 0.00	0.01 ± 0.01	OOM	0.00 ± 0.00	0.03 ± 0.01	0.01 ± 0.01	$\mathbf{0.54 \pm 0.11}$
	F1 Nodes↑	0.00 ± 0.00	0.01 ± 0.01	OOM	0.00 ± 0.00	0.01 ± 0.00	0.00 ± 0.01	$\mathbf{0.51 \pm 0.09}$
	Precision Old Nodes↑	0.00 ± 0.02	$\mathbf{0.57 \pm 0.45}$	OOM	0.00 ± 0.00	0.26 ± 0.07	0.09 ± 0.16	0.28 ± 0.07
	Recall Old Nodes↑	0.00 ± 0.00	0.02 ± 0.01	OOM	0.00 ± 0.00	0.06 ± 0.02	0.01 ± 0.02	$\mathbf{0.30 \pm 0.08}$
	F1 Old Nodes↑	0.00 ± 0.01	0.03 ± 0.02	OOM	0.00 ± 0.00	0.09 ± 0.03	0.02 ± 0.04	$\mathbf{0.28 \pm 0.06}$
	New Nodes Predicted↑	10.24 ± 7.16	0.79 ± 1.01	OOM	0.49 ± 0.71	7.82 ± 3.90	6.63 ± 6.09	$\mathbf{0.15 \pm 0.50}$
Structure Evaluation	Avg Node Degree→ 0	-0.37 ± 0.06	2.82 ± 0.72	OOM	3.41 ± 0.95	0.59 ± 1.16	-0.04 ± 0.15	$\mathbf{0.02 \pm 0.09}$
	Unique Degree Count→ 0	-0.53 ± 0.13	1.65 ± 0.84	OOM	1.89 ± 0.82	-0.24 ± 0.94	$\mathbf{0.07 \pm 0.35}$	0.10 ± 0.28
	Degree Centrality→ 0	-0.83 ± 0.14	5.28 ± 4.95	OOM	7.30 ± 5.43	-0.48 ± 0.56	-0.61 ± 0.26	$\mathbf{0.07 \pm 0.81}$
	Assortativity Coefficient→ 0	-2.26 ± 0.49	$\mathbf{0.06 \pm 0.37}$	OOM	0.92 ± 0.55	1.59 ± 0.93	-1.66 ± 0.81	0.61 ± 0.52
	Clustering Coefficient→ 0	$\mathbf{1.36 \pm 3.48}$	19.54 ± 34.79	OOM	25.50 ± 47.62	10.55 ± 29.54	7.60 ± 17.10	2.74 ± 7.26
	Density→ 0	-0.83 ± 0.14	5.28 ± 4.95	OOM	7.30 ± 5.43	-0.48 ± 0.56	-0.61 ± 0.26	$\mathbf{0.07 \pm 0.81}$
	Num Triangles→ 0	5.91 ± 18.95	448.38 ± 863.70	OOM	488.80 ± 1095.20	281.67 ± 1744.48	44.18 ± 110.32	$\mathbf{4.81 \pm 3.54}$
	Descriptor Norm→ 0	935.66 ± 592.46	222.01 ± 227.53	OOM	217.45 ± 219.85	658.49 ± 202.90	452.03 ± 566.43	$\mathbf{122.31 \pm 195.08}$
	Median Extra Nodes→ 0	237.00 ± 15.06	$\mathbf{0.00 \pm 0.00}$	OOM	$\mathbf{0.00 \pm 0.00}$	197.00 ± 7.56	129.00 ± 10.25	5.00 ± 2.85
	Median Missing Nodes→ 0	$\mathbf{0.00 \pm 0.00}$	23.00 ± 3.02	OOM	32.00 ± 2.32	$\mathbf{0.00 \pm 0.00}$	$\mathbf{0.00 \pm 0.00}$	0.00 ± 0.54
	Median Extra Edges→ 0	110.00 ± 8.97	96.00 ± 6.98	OOM	106.00 ± 8.67	267.00 ± 15.10	109.00 ± 8.44	$\mathbf{5.00 \pm 2.72}$
	Median Missing Edges→ 0	$\mathbf{0.00 \pm 0.00}$	$\mathbf{0.00 \pm 0.00}$	OOM	$\mathbf{0.00 \pm 0.00}$	$\mathbf{0.00 \pm 0.00}$	$\mathbf{0.00 \pm 0.00}$	0.00 ± 0.38

Table 23: Comprehensive evaluation results—including edge, node, and structural metrics—for all benchmark models on **Reddit B**. $\rightarrow 0$ indicates that values closer to 0 are better. $\uparrow$ denotes that higher values are better, while $\downarrow$ denotes that lower values are better. **Bold** indicates best, the second-best is underlined.

	Metric	HTGN	ROLAND	VGRNN	TGCN	GCLSTM	EvolveGCN	TopoGED
Edge Evaluation	oo-bank Precision↑	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	$\mathbf{0.05 \pm 0.03}$
	oo-bank Recall↑	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	$\mathbf{0.10 \pm 0.08}$
	oo-bank F1↑	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	$\mathbf{0.06 \pm 0.03}$
	oo-nobank Precision↑	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	$\mathbf{0.00 \pm 0.01}$
	oo-nobank Recall↑	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	$\mathbf{0.00 \pm 0.01}$
	oo-nobank F1↑	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	$\mathbf{0.00 \pm 0.01}$
	Edge Precision↑	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	$\mathbf{0.03 \pm 0.02}$
	Edge Recall↑	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	$\mathbf{0.03 \pm 0.02}$
	Edge F1↑	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	$\mathbf{0.03 \pm 0.02}$
	Num o-n Predicted→ 0	-1.00 ± 0.00	-1.00 ± 0.00	-1.00 ± 0.00	-1.00 ± 0.00	-1.00 ± 0.00	-1.00 ± 0.00	$\mathbf{0.11 \pm 0.51}$
	Num n-n Predicted→ 0	467.53 ± 284.41	461.78 ± 269.78	866.86 ± 528.87	466.79 ± 283.91	782.58 ± 470.41	466.82 ± 283.93	$\mathbf{0.59 \pm 1.56}$
Node Evaluation	Precision Nodes↑	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	$\mathbf{0.38 \pm 0.04}$
	Recall Nodes↑	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	$\mathbf{0.38 \pm 0.03}$
	F1 Nodes↑	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	$\mathbf{0.38 \pm 0.03}$
	Precision Old Nodes↑	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	$\mathbf{0.34 \pm 0.04}$
	Recall Old Nodes↑	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	$\mathbf{0.34 \pm 0.03}$
	F1 Old Nodes↑	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	$\mathbf{0.34 \pm 0.03}$
	New Nodes Predicted↑	23.99 ± 6.86	4.88 ± 2.47	3.82 ± 1.78	6.87 ± 5.81	3.22 ± 1.11	8.53 ± 4.13	$\mathbf{0.04 \pm 0.31}$
Structure Evaluation	Avg Node Degree→ 0	0.41 ± 0.16	5.55 ± 2.10	13.17 ± 2.96	5.09 ± 2.65	13.02 ± 1.71	3.01 ± 1.05	$\mathbf{0.02 \pm 0.08}$
	Unique Degree Count→ 0	1.65 ± 0.55	2.71 ± 0.86	2.13 ± 0.81	2.31 ± 1.14	4.63 ± 0.98	2.78 ± 0.90	$\mathbf{0.87 \pm 0.39}$
	Degree Centrality→ 0	-0.22 ± 0.15	17.19 ± 9.94	43.45 ± 16.60	16.15 ± 10.87	45.20 ± 7.95	5.70 ± 3.12	$\mathbf{0.03 \pm 0.11}$
	Assortativity Coefficient→ 0	-3.63 ± 28.88	6.25 ± 65.25	10.71 ± 77.63	8.28 ± 67.75	5.16 ± 60.15	$\mathbf{1.00 \pm 21.05}$	1.10 ± 32.12
	Clustering Coefficient→ 0	3.78 ± 13.51	44.47 ± 110.98	58.99 ± 156.80	42.99 ± 161.65	46.75 ± 133.31	18.76 ± 52.63	$\mathbf{2.07 \pm 6.25}$
	Density→ 0	-0.22 ± 0.15	17.19 ± 9.94	43.45 ± 16.60	16.15 ± 10.87	45.20 ± 7.95	5.70 ± 3.12	$\mathbf{0.03 \pm 0.11}$
	Num Triangles→ 0	151.71 ± 211.40	843.41 ± 1136.71	1707.68 ± 2023.11	687.13 ± 934.66	1970.28 ± 2612.84	669.92 ± 789.01	$\mathbf{23.25 \pm 38.07}$
	Descriptor Norm→ 0	1046.93 ± 276.60	1002.78 ± 148.13	1495.89 ± 206.22	1030.84 ± 319.63	1421.72 ± 174.75	755.22 ± 126.19	$\mathbf{446.58 \pm 67.84}$
	Median Extra Nodes→ 0	352.00 ± 16.18	4.00 ± 0.00	$\mathbf{0.00 \pm 0.00}$	269.00 ± 0.00	$\mathbf{0.00 \pm 0.00}$	71.00 ± 0.00	23.00 ± 4.24
	Median Missing Nodes→ 0	$\mathbf{0.00 \pm 0.00}$	253.00 ± 10.95	291.00 ± 6.25	265.00 ± 9.92	299.00 ± 8.01	161.00 ± 19.08	0.00 ± 2.24
	Median Extra Edges→ 0	463.00 ± 20.23	460.00 ± 20.39	1101.00 ± 21.94	462.00 ± 21.05	967.00 ± 17.34	462.00 ± 20.31	$\mathbf{13.00 \pm 7.77}$
	Median Missing Edges→ 0	$\mathbf{0.00 \pm 0.00}$	$\mathbf{0.00 \pm 0.00}$	$\mathbf{0.00 \pm 0.00}$	$\mathbf{0.00 \pm 0.00}$	$\mathbf{0.00 \pm 0.00}$	$\mathbf{0.00 \pm 0.00}$	0.00 ± 1.38

Table 24: Comprehensive evaluation results—including edge, node, and structural metrics—for all benchmark models on **TGBL-Wiki**. $\rightarrow 0$ indicates that values closer to 0 are better. $\uparrow$ denotes that higher values are better, while $\downarrow$ denotes that lower values are better. **Bold** indicates best, the second-best is underlined.

	Metric	HTGN	ROLAND	VGRNN	TGCN	GCLSTM	EvolveGCN	TopoGED
Edge Evaluation	oo-bank Precision↑	0.39 ± 0.23	0.24 ± 0.15	0.00 ± 0.00	$\mathbf{0.65 \pm 0.40}$	0.00 ± 0.00	0.00 ± 0.00	0.27 ± 0.16
	oo-bank Recall↑	0.03 ± 0.02	0.01 ± 0.01	0.00 ± 0.00	0.01 ± 0.01	0.00 ± 0.00	0.00 ± 0.00	$\mathbf{0.34 \pm 0.20}$
	oo-bank F1↑	0.05 ± 0.03	0.02 ± 0.01	0.00 ± 0.00	0.02 ± 0.01	0.00 ± 0.00	0.00 ± 0.00	$\mathbf{0.30 \pm 0.18}$
	oo-nobank Precision↑	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	$\mathbf{0.07 \pm 0.07}$
	oo-nobank Recall↑	0.01 ± 0.01	0.01 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	$\mathbf{0.07 \pm 0.08}$
	oo-nobank F1↑	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	$\mathbf{0.07 \pm 0.08}$
	Edge Precision↑	0.01 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	$\mathbf{0.17 \pm 0.01}$
	Edge Recall↑	0.01 ± 0.00	0.01 ± 0.00	0.00 ± 0.00	0.01 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	$\mathbf{0.18 \pm 0.01}$
	Edge F1↑	0.01 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	$\mathbf{0.18 \pm 0.01}$
	Num o-n Predicted→ 0	-0.89 ± 0.03	-1.00 ± 0.00	-1.00 ± 0.00	-0.94 ± 0.10	0.90 ± 0.24	-1.00 ± 0.00	$\mathbf{0.14 \pm 0.18}$
	Num n-n Predicted→ 0	$\mathbf{0.05 \pm 0.94}$	-0.60 ± 0.55	-0.60 ± 0.55	-0.60 ± 0.55	14.76 ± 12.79	-0.60 ± 0.55	3.09 ± 5.06
Node Evaluation	Precision Nodes↑	0.57 ± 0.03	0.33 ± 0.05	0.11 ± 0.01	0.34 ± 0.03	0.53 ± 0.09	0.20 ± 0.03	$\mathbf{0.63 \pm 0.03}$
	Recall Nodes↑	0.25 ± 0.01	0.18 ± 0.01	0.48 ± 0.03	0.13 ± 0.08	0.08 ± 0.01	0.18 ± 0.08	$\mathbf{0.67 \pm 0.02}$
	F1 Nodes↑	0.35 ± 0.01	0.24 ± 0.02	0.17 ± 0.01	0.17 ± 0.07	0.13 ± 0.02	0.17 ± 0.05	$\mathbf{0.65 \pm 0.01}$
	Precision Old Nodes↑	0.57 ± 0.03	0.33 ± 0.05	0.11 ± 0.01	0.34 ± 0.02	0.57 ± 0.06	0.20 ± 0.03	$\mathbf{0.60 \pm 0.02}$
	Recall Old Nodes↑	0.28 ± 0.01	0.21 ± 0.02	0.55 ± 0.04	0.15 ± 0.09	0.08 ± 0.01	0.20 ± 0.10	$\mathbf{0.63 \pm 0.02}$
	F1 Old Nodes↑	0.38 ± 0.01	0.25 ± 0.02	0.18 ± 0.01	0.19 ± 0.08	0.14 ± 0.01	0.19 ± 0.05	$\mathbf{0.62 \pm 0.01}$
	New Nodes Predicted↑	-0.98 ± 0.00	-1.00 ± 0.00	-1.00 ± 0.00	-0.99 ± 0.01	-0.86 ± 0.02	-1.00 ± 0.00	$\mathbf{0.14 \pm 0.18}$
Structure Evaluation	Avg Node Degree→ 0	5.13 ± 0.40	4.02 ± 0.63	0.14 ± 0.02	8.12 ± 4.85	17.51 ± 1.16	2.59 ± 2.12	$\mathbf{0.01 \pm 0.02}$
	Unique Degree Count→ 0	1.42 ± 0.20	2.24 ± 0.32	-0.90 ± 0.01	2.35 ± 0.40	2.92 ± 0.23	1.51 ± 0.55	$\mathbf{0.26 \pm 0.19}$
	Degree Centrality→ 0	13.11 ± 1.64	8.38 ± 2.43	-0.75 ± 0.02	38.96 ± 33.94	130.76 ± 13.01	5.27 ± 7.07	$\mathbf{-0.05 \pm 0.06}$
	Assortativity Coefficient→ 0	-6.90 ± 1.44	3.51 ± 1.32	8.97 ± 2.27	3.91 ± 0.48	4.27 ± 1.21	3.80 ± 1.48	$\mathbf{1.58 \pm 0.35}$
	Clustering Coefficient→ 0	0.36 ± 0.01	$\mathbf{0.00 \pm 0.00}$	$\mathbf{0.00 \pm 0.00}$	0.56 ± 0.20	0.79 ± 0.01	0.41 ± 0.28	0.02 ± 0.01
	Density→ 0	13.11 ± 1.64	8.38 ± 2.43	-0.75 ± 0.02	38.96 ± 33.94	130.76 ± 13.01	5.27 ± 7.07	$\mathbf{-0.05 \pm 0.06}$
	Num Triangles→ 0	11045.00 ± 1653.77	$\mathbf{0.00 \pm 0.00}$	$\mathbf{0.00 \pm 0.00}$	30045.00 ± 13822.67	40188.20 ± 6563.66	11052.40 ± 4986.92	99.60 ± 39.51
	Descriptor Norm→ 0	3447.24 ± 307.45	2997.41 ± 242.59	17861.99 ± 1923.61	3205.85 ± 759.27	3762.26 ± 296.79	2652.77 ± 461.59	$\mathbf{1034.67 \pm 135.03}$
	Median Extra Nodes→ 0	$\mathbf{0.00 \pm 0.00}$	$\mathbf{0.00 \pm 0.00}$	5104.00 ± 414.76	$\mathbf{0.00 \pm 0.00}$	$\mathbf{0.00 \pm 0.00}$	0.00 ± 203.32	99.00 ± 51.77
	Median Missing Nodes→ 0	753.00 ± 72.77	646.00 ± 77.90	$\mathbf{0.00 \pm 0.00}$	1008.00 ± 294.69	1154.00 ± 66.14	77.00 ± 257.93	0.00 ± 4.34
	Median Extra Edges→ 0	2123.00 ± 176.71	2257.00 ± 117.44	5292.00 ± 418.19	2070.00 ± 185.88	2070.00 ± 188.06	2070.00 ± 185.93	$\mathbf{87.00 \pm 60.57}$
	Median Missing Edges→ 0	$\mathbf{0.00 \pm 0.00}$	$\mathbf{0.00 \pm 0.00}$	$\mathbf{0.00 \pm 0.00}$	$\mathbf{0.00 \pm 0.00}$	$\mathbf{0.00 \pm 0.00}$	$\mathbf{0.00 \pm 0.00}$	0.00 ± 1.68

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction consistently present the work as an inverse-topology approach to temporal graph forecasting that models node churn and global structure. These claims are well aligned with the paper's contributions, scope, and empirical evaluation.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We discuss the limitation of our work at the end of Section 5.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: We provide Proposition 4.1 to support our claim that topology descriptors used in our method are stable forecasting targets. We provide the corresponding proof in Section D.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: The paper provides a comprehensive description of the datasets, model configurations, and hyperparameters used, ensuring that the experimental results are reproducible. A link to an Anonymous Repository containing code for reproducing is provided under Section 5.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: Code to reproduce results is publicly available at https://github.com/FDataLab/TopoGED, with detailed instructions provided in the supplemental material to ensure reproducibility.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Choices of hyperparameters (including optimizer, scheduler, etc.) for each model used in our paper can be found in Section 5. For each dataset, data splits can be retrieved from running the code at the Anonymous Repository (https://github.com/FDataLab/TopoGED/blob/main/README.md)

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

Justification: We show results of three runs along with standard deviations, providing clear information on the statistical significance of the experiments.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: The paper details the compute resources used for the experiments, including the type of hardware, memory, and time required for execution, ensuring transparency and reproducibility, Section 5.

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?

   Answer: [Yes]

   Justification: The authors have read and confirmed that the research adheres to the NeurIPS Code of Ethics, with considerations for responsible AI practices and transparency in the methodologies and data used.

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: Broader Impact is discussed in Secion A

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The paper does not involve the release of models or data that have a high risk of misuse, and thus this question is not applicable.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: The paper credits the creators of all assets used, including datasets and code, and adheres to the respective licenses and terms of use.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: We do not introduce any new assets in this paper.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing or research with human subjects.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [N/A]

Justification: The research does not involve human subjects, and therefore, IRB approval is not applicable.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [N/A]

Justification: The research doesn't involve LLMs