

Pair-Level Candidate Generation for Threat Early Warning over Temporal Security Knowledge Graphs

Jingtao Peng

Southwest Minzu University, Chengdu 610225, China

Abstract: Threat early warning must flag suspicious actor interactions from open-source intelligence ahead of time. We frame it as temporal link prediction over a temporal security knowledge graph, whose output is an early-warning candidate for human triage. TGN-Sec adapts the temporal graph network to the security setting via a relation-aware message function (Modification A) and temporal decay attention (Modification B), mapping scores to candidates through relative ranking; online maintenance loops via memory updates, elastic weight consolidation (EWC) fine-tuning, and a Kolmogorov–Smirnov (KS) test for drift retraining. Built from GDELT 2.0, the evaluation graph spans 92 days, 167 entities, 14,223 edges, and 5 CAMEO-derived relation types; under an aligned protocol we ablate the two modifications, compare against five baselines, and run a comparability check exposing 8 defects. On pair-level early-warning candidate generation TGN-Sec reaches F1 of 0.1074 ± 0.0134 against the top baseline TransE at 0.0402, about 2.2 to 3.0 times, with pair-level recall leading 2.87 to 4.65 times and the lowest repeated-alert rate (29.99%), scoped to human triage rather than automatic alerting. The reverse also holds: item-level precision of 0.0762 ± 0.0090 lags the baselines (0.2427 to 0.2813) by 3.2 to 3.7 times, the two modifications show no significant independent contribution, and recall on pairs unseen in training is 0.000 across three seeds. This paper favors honest assessment of a constructed system and treats protocol alignment as a prerequisite before any continuous-time metric is reported.

Keywords: Temporal Security Knowledge Graph, Temporal Link Prediction, Early-Warning Candidate Generation, Temporal Graph Network, Concept Drift.

1. Introduction

Geopolitical tensions surface through public signals. Sanctions, military exercises, and statements of position move between states and reshape the threat environment well before any confrontation becomes explicit, and the same interactions transmit along the relations that connect actors. Single weak signals carry limited meaning on their own; several of them travelling along the actor network form a readable posture only in aggregate. Human analysts who follow these streams face two pressures at once: the volume of open-source reporting outruns manual review, and the interval between an early interaction and a visible event is short. Zhang et al. frame threat early warning as a task over a security knowledge graph built from threat intelligence [1]. We adopt the same stance and cast early warning as temporal link prediction over a temporal security knowledge graph, in which a future edge between two actors is a candidate threat relationship to be ranked ahead of time.

This paper contributes along three lines: design, evaluation, and finding, with an honest assessment of a constructed system rather than a claim of method advance.

We first adapt the temporal graph network (TGN) to the security setting. A relation-aware message function (Modification A) places the relation type directly into message generation, so that sanction, conflict, and cooperation are not collapsed to a single edge label, and temporal decay attention (Modification B) weighs recent neighbours above older ones at readout. The model maps link-prediction scores to early-warning candidates through relative ranking: a candidate is decided by its position among all pairs, not by an absolute probability threshold, which keeps the output comparable across memory states.

The evaluation runs on a temporal security knowledge graph built from the GDELT 2.0 event database, covering 92

days over 167 entities, 14,223 edges, and 5 CAMEO-derived relation types. Under an aligned protocol we ablate the two modifications, compare TGN-Sec against five baselines (TransE, LSTM-Sequence, TGN, TGAT, DyRep), and carry out a protocol-comparability check. Pairing a same-source precision-recall examination with multi-protocol cross-checks, that check exposed 8 defects that would have shifted the reported numbers.

The central empirical result is a lead on pair-level early-warning candidate generation. TGN-Sec reaches pair-level F1 of 0.1074 ± 0.0134 against TransE at 0.0402, the top baseline on pair-level F1, a factor of about 2.2 to 3.0 times, with a pair-level recall lead of 2.87 to 4.65 times and the lowest repeated-alert rate in the comparison (29.99%). This lead is scoped to candidate generation for human triage, not to automatic alerting. The same evaluation reports the reverse direction: on item-level precision TGN-Sec (0.0762 ± 0.0090) lags the baselines (0.2427 to 0.2813) by 3.2 to 3.7 times. The two modifications show no significant independent contribution, the four ablation groups all fall within the within-group standard deviation, with paired $|t|$ at most 3.08 ($df = 2$, critical 4.303), and recall on entity pairs unseen during training is 0.000 across three seeds. Reporting negative results plainly is the orientation of this work.

The remainder of this paper is organized as follows. Section 2 reviews temporal graph models, temporal link prediction, and security knowledge graphs. Section 3 formalizes the temporal security knowledge graph and the warning task. Section 4 presents TGN-Sec and its online maintenance. Section 5 reports the experiments, and Section 6 discusses the negative results and their implications. Section 7 concludes.

2. Related Work

(1) Temporal graph models

Temporal graph networks update node representations as interactions arrive, rather than over a fixed graph. The designs differ in how they carry time. TGAT [2] turns each timestamp into a vector and uses a self-attention layer over temporal neighbours, so a node's embedding moves with the interaction stream without discretising time into windows. TGN instead keeps a memory module that a message function and a recurrent aggregator refresh at every event, then reads it out for downstream scoring [3]. The same pipeline supports forecasting: Nayeri and Rezvani cast alert prediction as finding the next victim with a transformer-based temporal graph network [4], and Dudziak-Gajowiak apply the TGN structure to link prediction [5]. Later work widens the view of time. DyRep [6] splits a node's growth from its communication through dual time scales, and Alilou et al. argue that temporal motifs carry signal beyond pairwise edges [7]. These models nonetheless share one trait: a single message function treats the relation type as a uniform label, and the relation enters only as an embedding slot, flattening sanction, conflict and cooperation to the same vector. That loss matters for security, where the three relations push an endpoint's threat state in different directions.

(2) Temporal link prediction

Temporal link prediction asks which edges appear next on a dynamic graph, and it has become a standard testbed for temporal graph models [5]. The task is framed as ranking a positive edge against sampled negatives at each timestamp, and progress is tracked through reciprocal rank and hit-rate metrics [8]. A unified theoretical and deep-learning framework consolidates these formulations and clarifies how training, negative sampling and evaluation interact [8]. Within this framing, the evaluation horizon and the negative set decide what the metrics mean: a short horizon rewards memorisation, whereas a long horizon under strict temporal extrapolation exposes generalisation. Reported conclusions shift with the evaluation protocol, because the candidate set size, the causal exclusion of future edges and the train/test time split all move the ranking. Most studies still report under heterogeneous protocols, so comparisons across papers are hard to reconcile and the ranking numbers are not directly comparable across models; a method that looks strong under one split may therefore not hold under another. The same instability matters for security early warning, where a fixed horizon and a stable protocol determine whether a candidate list can be trusted for human triage.

(3) Security knowledge graphs and threat intelligence

Security knowledge graphs organise threat entities and relations into a multi-relational structure for reasoning. Unlike general knowledge graphs, the relation set carries threat semantics that change how an edge should be read. Ren et al. build CSKG4APT to support attribution of advanced persistent threat actors [9], and Zhang et al. propose Warning-Graph, an early-warning framework that models threat intelligence for APT detection [1]. Ge et al. pursue robust threat attribution under evolving attacks, treating the relation as part of the propagation semantics [10]. Beyond graph construction, online detection must cope with non-stationary data: De Paola et al. present HOIDS, a concept-drift-aware online intrusion detection system that triggers adaptation when the distribution shifts [11]. These efforts establish the security knowledge graph and the need for drift-aware online

operation. Yet they rarely model threat propagation as a continuous-time graph process, remain offline or batch oriented in attribution and detection, and seldom connect the security graph to a forward-looking warning output that an analyst can act on through triage.

Existing temporal graph models, including TGN-family methods, pass the relation type into a single message function as a uniform label [3], yet security relations such as sanction, conflict and cooperation carry non-equivalent threat semantics [10]. These models report results mainly with ranking metrics, while security early warning depends on whether, under a strict protocol that covers full candidates, temporal extrapolation and a fixed horizon, the model can steadily produce candidates for human triage and whether it can predict entity pairs unseen in training [8]. This paper therefore centres evaluation on the pair-level outcome of early-warning candidates and on the boundary of predictability, and reports comparisons with baselines without claiming an a priori advantage.

3. Problem Formulation

(1) Temporal security knowledge graph

A temporal security knowledge graph is a tuple $G = (V, E, T)$. The set V contains the entity nodes, E contains the timestamped interaction edges, and T is the time axis along which the interactions are ordered. Every edge is a five-tuple $e = (s, d, r, t, w)$, where $s \in V$ is the source node, $d \in V$ is the target node, r is the relation type drawn from a finite relation set R , $t \in T$ is the timestamp of occurrence, and w is the interaction strength. R comprises five CAMEO-derived relation types: cooperation, conflict, sanction, statement, and military exercise. Edge direction follows the actor ordering of the underlying event record, so (s, d) and (d, s) are distinct.

The evidence available at a cut-off time t is the history observed no later than t , which makes the setting causal:

$$H_t = \{ (s, d, r, t') \in E : t' \leq t \} \quad (1)$$

(2) Task definition

Temporal link prediction asks which interactions come next. Given the history of Eq. (1), the task is to decide, for every ordered node pair and relation type, whether an edge appears inside the horizon window $(t, t + \Delta]$, where Δ (denoted H in the experimental tables) is the horizon. The task amounts to learning a parameterised scoring function

$$\hat{y}(s, d, r | H_t) = f_{\theta}(s, d, r, H_t) \quad (2)$$

Where θ collects the model parameters, $\hat{y}$ is the score of the triple (s, d, r) within the horizon, and the predicted probability is $\sigma(\hat{y})$.

The candidate space $C = V \times V \times R$ holds 167 source nodes, 166 targets per source once self-pairs are removed, and 5 relation types, so $|C| = 167 \times 166 \times 5 = 138,610$. A warning issued at time t is the TopK-scored subset of C ,

$$A_t = \text{TopK}_{((s, d, r) \in C)} \hat{y}(s, d, r | H_t),$$

Where A_t is the warning set issued at t , TopK returns the top_k triples under descending score, and Section 4.5 fixes both values.

The temporal ordering fixes a 0.7 / 0.15 / 0.15 partition across training, validation and test, with every test interaction later than all training ones so that no future information contaminates the fit.

The formulation is an instance of representation learning

on continuous-time dynamic graphs [7] and aligns with a unified framework for temporal link prediction in dynamic networks [8]. Beyond this shared framing, two traits set the present task apart: each edge is tagged with a security relation type, and the prediction target is associated with a lead time.

4. Method

(1) Model overview

TGN-Sec builds on the temporal graph network (TGN) [3]

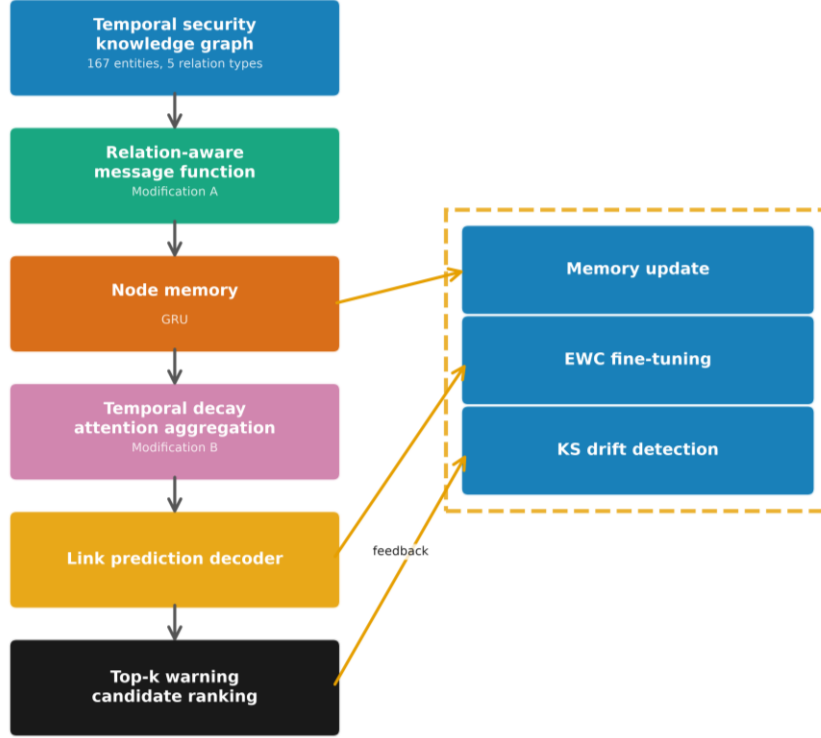


Figure 1. Overview of TGN-Sec: message generation, node memory update, temporal decay attention aggregation, and link prediction decoding, followed by the online maintenance loop

(2) Relation-aware message function (Modification A)

Different relation types move the endpoint threat state by unequal magnitude and direction, so a single edge label cannot preserve that asymmetry once the message is formed. Modification A gives the relation type a direct path into message generation, as in [10]:

$$m_{(s \rightarrow d)}(t) = \text{LayerNorm}(W^{(2)} \cdot \text{ReLU}(W^{(1)} [h_s; h_d; e_r; \phi(t)] + b^{(1)}) + b^{(2)}) \quad (3)$$

Where $h_s, h_d \in \mathbb{R}^d$ are the endpoint embeddings, $e_r \in \mathbb{R}^d$ is the learnable embedding of relation r , $\phi(t) \in \mathbb{R}^d$ is the sinusoidal timestamp encoding, $[\cdot; \cdot]$ is concatenation, $W^{(1)}, W^{(2)}, b^{(1)}, b^{(2)}$ are perceptron parameters, and LayerNorm is layer normalisation. With $d = 128$ and $d_t = 128$, the input has $3d + d_t = 512$ entries.

The message of Eq. (3) updates both endpoints with the head and the tail swapped: $m_{(s \rightarrow d)}(t)$ writes to the memory of d , $m_{(d \rightarrow s)}(t)$ to the memory of s .

Placing e_r alongside the endpoint and time encodings lets the first perceptron layer combine the relation signal through an unconstrained linear map, so no dedicated gate silently suppresses it.

(3) Temporal decay attention (Modification B)

Threat does not stay at one node; it cascades along the graph, so a readout that consults only the node's own memory throws away the neighbour history that carries the threat forward. Modification B aggregates the recent temporal

and carries out its three computational stages in turn: a message function forms a message from the two endpoint memories, the relation type and the timestamp; a memory module — a gated recurrent unit (GRU) — rewrites each node's memory from its prior value and the aggregated message; and a neighbour aggregation merges recent-neighbour states with the static node embedding before the decoder produces the score of Eq. (2). Fig. 1 shows the data flow, and Section 4.6 describes the maintenance loop.

neighbours at readout: for a node s at time t , the $k = 10$ most recent interactions supply the neighbour memories, and the lookup keeps only timestamps strictly earlier than t .

For a neighbour interaction observed at t' , the elapsed time is

$$\Delta t = t - t' \quad (4)$$

And its weight decays exponentially:

$$\alpha(\Delta t) = \exp(-\lambda \cdot \Delta t) \quad (5)$$

Where λ is the decay coefficient, set to $\lambda = 0.01$, which gives a half-life of $\ln 2 / \lambda$, about 69 days.

Aggregation uses multi-head attention with 2 heads [4]: the memory $s_s(t)$ is the query, the neighbour memories are the keys and the values, and the decay weight of Eq. (5) scales the dot product:

$$\text{score}_{(s,j)}(h) = ((Q_h s_s(t)) \cdot (K_h s_j(t'))) / \sqrt{(d/H) \cdot \alpha(t - t')} \quad (6)$$

Where Q_h, K_h and V_h are the linear transformations of head h , $s_j(t')$ is the memory of neighbour j , and H is the number of heads. A softmax over neighbours normalises the scores of Eq. (6), the values are combined by a weighted sum, and the heads are concatenated and projected; a residual connection and layer normalisation follow, giving h_{agg} . A combination layer fuses h_{agg} with the static node embedding:

$$z_s(t) = \text{ReLU}(W_c [h_{agg}; h_s]) \quad (7)$$

Where W_c is the projection of the combination layer. Because neighbours contribute unequally, a plain mean pool would blur their signal; instead the attention weights them by content relevance, while the decay term supplies the missing time prior.

(4) Link prediction decoding and training loss

The decoder scores a candidate triple from the representations of its two endpoints and the relation embedding, using three interaction features: the concatenation of the endpoint representations, their element-wise product, and their absolute difference. The decoder input is

$$\text{score}(s, d, r) = \text{MLP}([z_s; z_d; z_s \odot z_d; |z_s - z_d|; e_r]) \quad (8)$$

Where $\odot$ is the element-wise product, $|\cdot|$ is the absolute value, and MLP is a multilayer perceptron. The decoder must see the relation signal directly; routing it through the message and memory alone dilutes the distinction between relations, leaving different edges of one node pair with nearly indistinguishable scores.

The decoder of Eq. (8) returns logits rather than probabilities; it takes the combined node representation $z_s(t)$ of Eq. (7) and the relation embedding as inputs, and training minimises the loss

$$L = -[y \cdot \log \sigma(\text{score}^+) + (1 - y) \cdot \log \sigma(\text{score}^-)] \quad (9)$$

Where $\sigma(\cdot)$ is the sigmoid function, score^+ and score^- are the logits of a positive and a negative sample, and $y = 1$ for a positive sample. The positive-to-negative ratio is 1:1.

Under streaming causal exclusion, the sampler maintains the set of outgoing edges already consumed and draws negatives from entities beyond it, retrying up to 100 times; because the filter admits only past interactions, a sampled negative can never coincide with a true edge of the training period.

Evaluation uses full ranking, so every positive is ranked against all 167 candidate entities, and mean reciprocal rank and Hits@K follow from that rank.

(5) From scores to early-warning candidates

The batching interface scores all candidates against the current node memory without updating it per candidate, which makes absolute probabilities shift between memory states and hence incomparable; warning therefore rests on relative ranking rather than an absolute probability threshold, and the ranking position alone decides whether a candidate is issued. The warning set of Section 3.2 is therefore instantiated as

$$A_t = \{ (s, d, r) \in C : \text{rank}(\hat{y}(s, d, r | H_t)) \leq \text{top}_k \} \quad (10)$$

Where rank returns the descending position of the score inside C , $\text{top}_k = 100$, and the warning horizon is $\Delta = 7$ days.

A node's warning level scales with its graph importance. For a node i ,

$$I(i) = w_{\text{deg}} \cdot \text{deg}(i) + w_{\text{att}} \cdot \bar{a}(i) \quad (11)$$

Where $\text{deg}(i)$ is the degree of node i , $\bar{a}(i)$ is the mean attention weight of i over the interactions it takes part in, and w_{deg} and w_{att} weight the two terms. In Eq. (11) degree characterises how active a node is, attention captures how

strongly it transmits influence, and the two together push candidates of prominent nodes to a higher warning level.

The lead time measures how early a warning arrives. For an edge e that occurs at $t_{\text{event}}(e)$, the lead time is

$$LT(e) = t_{\text{event}}(e) - t_{\text{first}}(e) \quad (12)$$

Where $t_{\text{first}}(e)$ marks when e first joins the warning set, no later than the time it occurs. An edge that misses the warning window before occurring is treated as a miss, and the lead time of Eq. (12) is averaged in days across the hits. Read this way, the pair-level output is a ranked candidate set for human triage, not an automatic alert; analysts, not the model, decide which candidates to pursue.

(6) Online maintenance

After deployment the model maintains its state from the daily event stream. In the online phase the weights stay frozen and only the new edges advance the node memory; an elastic weight consolidation (EWC) term [12] guards against forgetting during periodic fine-tuning, and a Kolmogorov–Smirnov (KS) test on the prediction distribution [11] fires retraining once concept drift appears. We implement and evaluate the full loop; it shows no clear advantage over the frozen baseline (see Section 5 and Section 6).

5. Experiments

(1) Data and setup

We build the temporal security knowledge graph from the GDELT 2.0 event database [13], covering 2026-06-01 to 2026-08-31 (92 days) at four snapshots per day. The raw feed yields 453,106 event rows, which reduce to 56,880 triples, 14,480 after cleaning, and 14,223 edges after a time cut that drops 257 boundary edges, over 167 entities and 5 relation types. Relation labels follow the CAMEO root-code mapping [14]. The five relations are imbalanced: cooperation 10,832 edges (about 76%), conflict 1,790, sanction 1,193, statement 281, military exercise 127. Entities are restricted to countries and intergovernmental organizations.

The main model is split by time into train, validation, and test at 0.7 / 0.15 / 0.15, giving 9,956 / 2,133 / 2,134 edges; the baselines use a different split, 10,136 / 2,172 / 2,172, a difference that is stated explicitly here. Hyperparameters are fixed (Table 1): learning rate 1e-3, batch 512, 100 epochs with early stopping on validation MRR (patience 5), dimension 128, two heads, $\lambda = 0.01$, 1:1 training negatives, full-ranking evaluation over 167 candidates, three seeds (42, 123, 2024) with PYTHONHASHSEED = 0, relation-cap off.

Two training protocols are in play and must not be conflated. Ours is fix2: streaming causal-exclusion negative sampling with a retry limit of 100, and Modification B wired into neighbour aggregation. The baselines use protocol v2: a train-split exclusion set, 1-vs-all over 167 candidates, strict-greater ranking, and paired AUC. Their ranking metrics are therefore not placed side by side without this declaration. The two subsections also draw on different weight batches: Section 5.2 and the ablation use the 2026-09-09 weights, whereas Section 5.3 uses the 2026-09-16 retrained weights. Evaluation is aligned throughout by full ranking, by replaying training edges into node memory before scoring, and by same-seed paired comparison.

The dataset statistics and the shared configuration are summarised in Table 1.

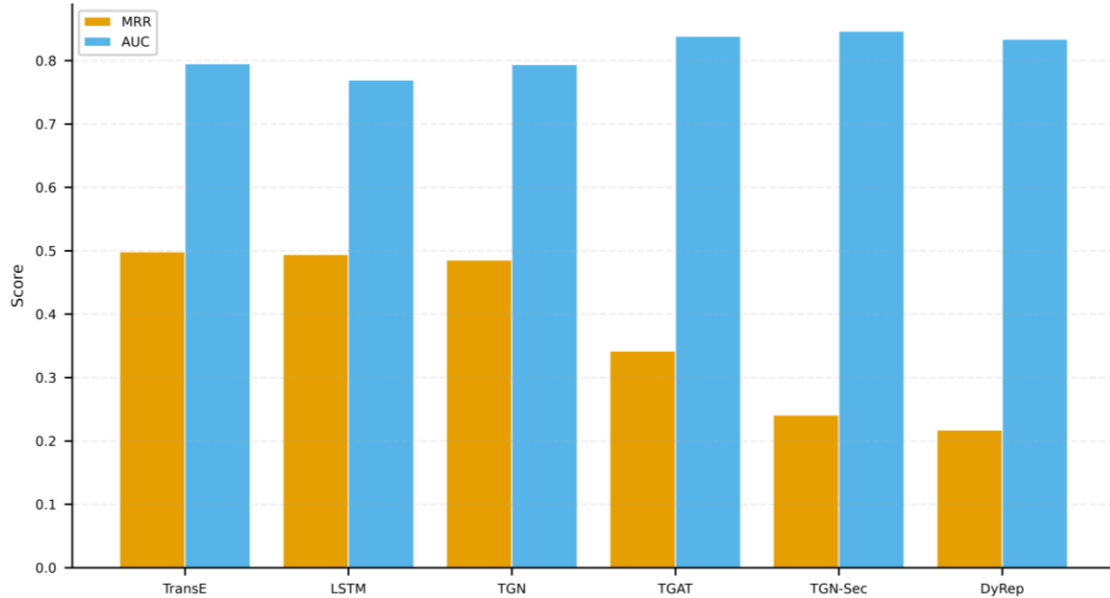
Table 1. Dataset statistics and the shared configuration.

Item	Value
Source	GDEL2 2.0, 2026-06-01 to 2026-08-31 (92 days) [13]
Raw events / triples / cleaned	453,106 / 56,880 / 14,480
Modelled edges / entities / relations	14,223 / 167 / 5
Relation split (coop / conf / sanc / stat / exer)	10,832 / 1,790 / 1,193 / 281 / 127
Main split (train / val / test)	9,956 / 2,133 / 2,134
Baseline split (train / val / test)	10,136 / 2,172 / 2,172
LR / batch / epochs (early stop)	1e-3 / 512 / 100 (val MRR, patience 5)
Dim / heads / lambda	128 / 2 / 0.01
Negatives (train / eval)	1:1 / full ranking over 167
Seeds / PYTHONHASHSEED	42, 123, 2024 / 0
Relation cap	off
Our protocol / baseline protocol	fix2 / v2

(2) Link prediction performance

Table 2 reports MRR, Hits@K (K = 1, 5, 10), and AUC for six models under the aligned protocol. The static embedding TransE [15] reaches the top MRR at 0.4984, followed by LSTM-Sequence [16] (0.4938), TGN [3] (0.4851), and TGAT [2] (0.3417); TGN-Sec scores 0.2407 and DyRep [6] 0.2169.

TGN-Sec therefore ranks fifth on MRR, roughly one half of TransE, and this gap is reported plainly rather than smoothed over. The one metric where TGN-Sec leads is AUC: its 0.8465 is the highest among the six, above TGAT (0.8383) and DyRep (0.8335) (Fig. 2).

**Figure 2.** Comparison of temporal link prediction performance (MRR and AUC) across six models on GDEL2; TGN-Sec attains the highest AUC

The split between a low MRR and a top AUC is not a contradiction. The model output is conservative: the final training loss is about 1.7 against a baseline range of 0.09 to 0.68, the negative tail of the score is not pushed down, and the positive probability settles near 0.7 versus roughly 0.95

for the baselines. AUC reads the whole score distribution and stays high, while MRR depends on placing the true target first and suffers. The same conservative calibration is discussed in Section 6.1 and is not repeated here.

Table 2. Temporal link prediction on GDEL2 (TGN-Sec is 3-seed mean ± std; baselines single seed).

Model	MRR	Hits@1	Hits@5	Hits@10	AUC
TransE	0.4984	0.4351	0.5493	0.6234	0.7949
LSTM-Sequence	0.4938	0.4406	0.5272	0.6082	0.7691
TGN	0.4851	0.4365	0.5129	0.5999	0.7937
TGAT	0.3417	0.2555	0.4042	0.5336	0.8383
TGN-Sec (ours)	0.2407 ± 0.0015	0.1204	0.3529	0.5152	0.8465 ± 0.0039
DyRep	0.2169	0.1142	0.2956	0.4724	0.8335

Three notes attach to the table. First, the training protocols differ: the baselines use a train-split exclusion set while TGN-Sec uses a streaming causal-exclusion set, so the two sets of

ranking numbers are not directly comparable. Second, the standard deviations are within-batch: another independent batch of the same configuration gives full-group MRR 0.2382

± 0.0075 (cross-batch spread about ± 0.01), and the six rounds merged across both batches give 0.2395 ± 0.0057 ; under either batch the four ablation groups show no significant difference. Third, the baselines are single-seed.

(3) Early-warning evaluation

Under the rolling protocol, each of 30 slices ($t = 213$ to 242 , aligned with the baselines) emits the $\text{top}_k = 100$ candidates

under a 7-day horizon over the full candidate space of 138,610, for 3,000 issued candidates in total. The 30 slices contain 1,240 deduplicated true events over 1,021 true entity pairs. Table 3 reports the unique-pair caliber, where pair-level precision is hits on unique pairs over warned unique pairs, recall is hits on unique pairs over 1,021, and F1 merges them from the same numerator, so the F1 is legitimate.

Table 3. Early-warning on GDELT (unique-pair caliber; TGN-Sec is 3-seed mean $\pm$ std, baselines single seed).

Model	Hit unique	Warned unique	Pair P	Pair R	Pair F1	Repeat %	Item P
TGN-Sec (ours)	60.3 ± 8.1	101.3 ± 11.1	0.5941 ± 0.0272	0.0591 ± 0.0079	0.1074 ± 0.0134	29.99 ± 3.51	0.0762 ± 0.0090
TransE	21	24	0.8750	0.0206	0.0402	125.00	0.2703
LSTM-Sequence	19	20	0.9500	0.0186	0.0365	150.00	0.2427
TGAT	18	20	0.9000	0.0176	0.0346	150.00	0.2627
TGN	14	20	0.7000	0.0137	0.0269	150.00	0.2813
DyRep	13	41	0.3171	0.0127	0.0245	73.17	0.0417

Reading the table, TGN-Sec leads the baselines on pair-

level F1 (Fig. 3; Table 3).

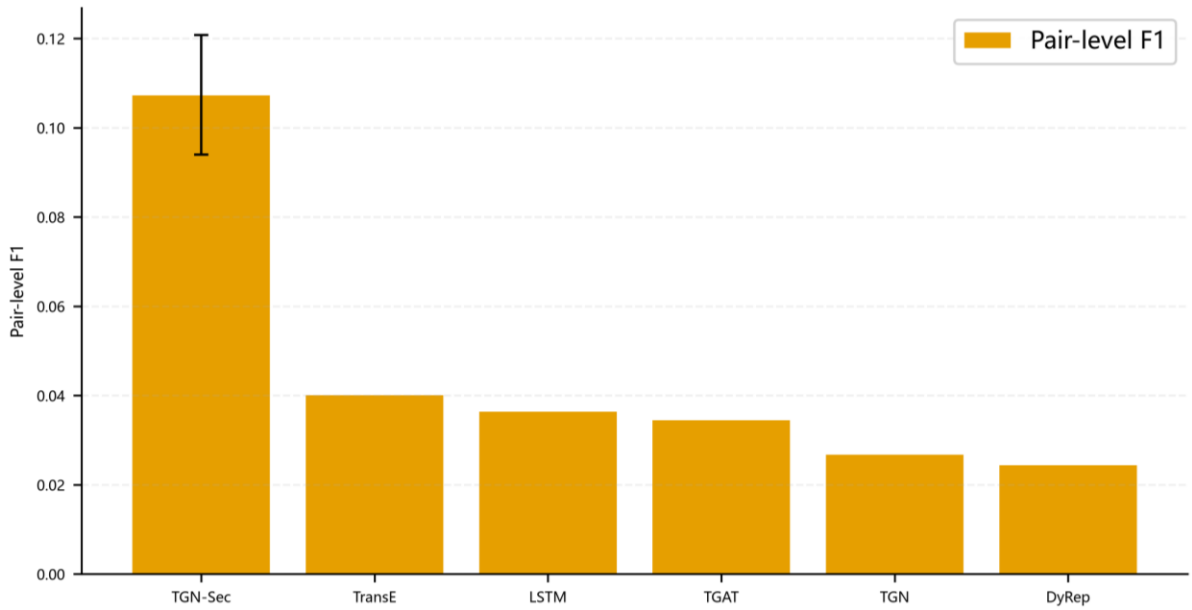


Figure 3. Pair-level warning F1 across six models on GDELT; TGN-Sec is reported as the mean and standard deviation over three seeds, whereas baselines are single-seed results. Its F1 is 0.1074 ± 0.0134 , against TransE at 0.0402, a factor of 2.67 (interval 2.34 to 3.01); under the alternative per-slice item caliber the lead is 2.48 times, and the two directions agree, so we state the lead as about 2.2 to 3.0 times. The pair-level recall lead is 2.87 to 4.65 times, and the repeated-alert rate of $29.99\% \pm 3.51\%$ is the lowest in the table. This is the only performance lead the paper claims, scoped to candidate generation for human triage, not automatic alerting.

The same table must show the reverse fact. The item-level precision of TGN-Sec is 0.0762 ± 0.0090 , while the baselines sit at 0.2427 to 0.2813; TGN-Sec therefore lags by 3.2 to 3.7 times on this metric. Reporting only the F1 lead would be selective. At the event level, item-level precision is 0.0762 ± 0.0090 (228.7 $\pm$ 26.8 hit items over 3,000 issued) and event-level recall is 0.0586 ± 0.0079 (72.7 $\pm$ 9.9 unique events over 1,240); we do not report an F1 at the event level, because the precision and recall numerators are not drawn from the same source. The lead time of hit events has mean 3.28 days, median 3.17, P25 1.50, and P75 4.94, where each quantile is the arithmetic mean of the three seeds' own quantiles.

Stratifying by training exposure, the 684 pairs seen during training yield 60.3 ± 8.1 hits (recall 0.088 ± 0.012), whereas

the 337 pairs never seen during training are missed by all three seeds.

The two panels of Fig. 4 frame the operating point. In panel (a), pair-level F1 rises monotonically with top_k with a slowing but non-vanishing gain (0.0324 at 20 to 0.2382 at 500; beyond 200 the gain holds at 0.020 to 0.025 per 100 points), while pair-level precision peaks at $\text{top}_k = 50$ (0.6812). We set the working point $\text{top}_k = 100$ as the analyst triage budget of 100 candidates per slice, not as an optimum. In panel (b), recall and F1 rise as the monitored-entity budget K shrinks, but only because the recall denominator shrinks with K (190 / 517 / 865 / 1,021 true pairs; event coverage 23.3% / 56.4% / 86.7% / 100% over 1,240). The panel reads as a coverage-cost trade-off, and F1 is not compared across K .

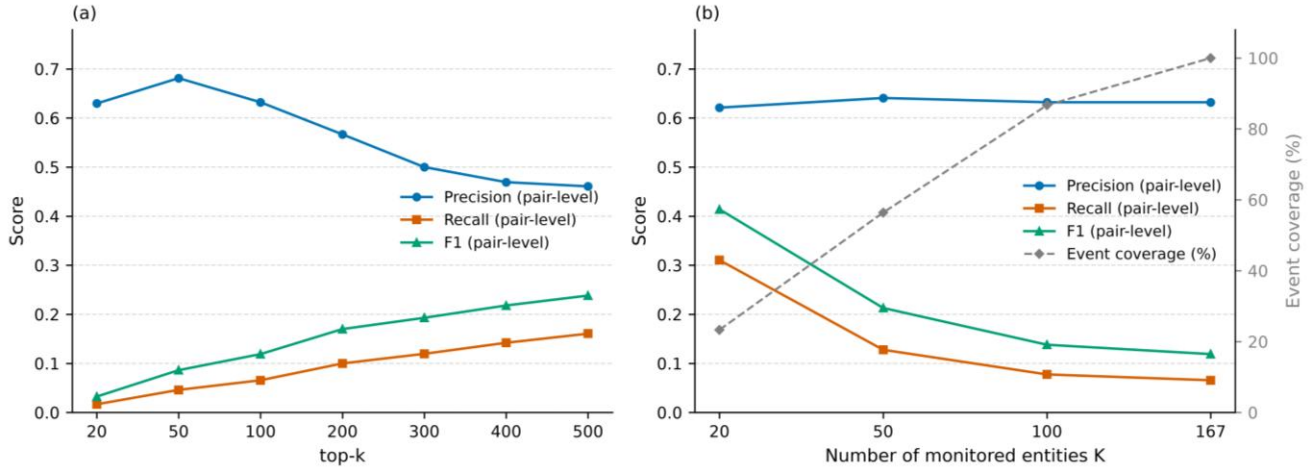


Figure 4. Pair-level precision, recall, and F1 of warning candidates under different top-k budgets (a) and different monitored-entity budgets K (b). In (a) F1 increases monotonically with a slowing but non-vanishing marginal gain, so the operating point top-k = 100 is set by the analyst triage budget rather than by an optimum; in (b) the recall denominator shrinks with K (190 / 517 / 865 / 1,021 true pairs) together with the event coverage (23.3% / 56.4% / 86.7% / 100%), so F1 is not comparable across K and the panel reads as a coverage-cost trade-off.

Table note: the baselines are single-seed without variance; TGN-Sec is the 3-seed mean $\pm$ std.

6. Discussion

(1) What the negative results show

The ablation study (Table 4) closes Modification A by

zeroing the relation embedding and Modification B by fixing the decay weight to one. At the current data scale neither modification shows a significant independent contribution; this is a statement about detectability, not that the mechanisms are inert. Across the four configurations the differences stay within the within-group standard deviation, and paired t-tests give $|t|$ at most 3.08 (df = 2, critical 4.303).

Table 4. Ablation over four configurations (3 seeds each; mean $\pm$ standard deviation).

Configuration	MRR	Hits@10	AUC
full	0.2407 $\pm$ 0.0015	0.5152	0.8465 $\pm$ 0.0039
noA	0.2426 $\pm$ 0.0021	0.5095	0.8528 $\pm$ 0.0026
noB	0.2434 $\pm$ 0.0022	0.5242	0.8480 $\pm$ 0.0032
bare	0.2457 $\pm$ 0.0056	0.5211	0.8498 $\pm$ 0.0017

Standard deviations are within-batch; the cross-batch comparison is given in Table 2.

The fault lies in the sampling protocol, not in the model. Under the earlier scheme about 29% of sampled negatives were known true training edges, giving contradictory supervision. Streaming causal exclusion with a higher retry limit collapsed the seed-to-seed MRR spread from 0.2675 to 0.0139, a near nineteen-fold variance cut; the collapse arrived without any change to the optimizer or capacity, which pins the fault on the sampling protocol.

Predictability is also bounded by the pairs the data supports. Recall by entity pair is 0.088 ± 0.012 (three seeds) for pairs seen in training and 0.000 for unseen ones, so no unseen pair was hit in any seed and the model anticipates existing interactions while missing exogenous ones. This matches the ablation null: the extractable signal is a first-order prior (who is active, who recently interacted), not a deeper temporal pattern.

The conservative output has a signature. The reason AUC ranks first while MRR trails is the calibration itself: final loss sits around 1.7 against a baseline range of 0.09 to 0.68, the negative tail is not pushed down, and the positive probability settles near 0.7 versus roughly 0.95 for the baselines. High AUC and low MRR are not contradictory; they describe the same conservative calibration from two angles.

The online maintenance follows the same logic. Across the three arms—static, memory, finetune—slice-wise metrics are essentially identical: finetuning does not beat the frozen static arm and EWC only prevents collapse. The shared positive

slope reflects the data's temporal progression, not the model update. New-pair recall is zero, so the absent online gain shares its origin with the ablation null.

(2) Implications for evaluation and deployment

The training protocols are not interchangeable: this work uses a streaming causal-exclusion set and the baselines a train-split set, so their ranking metrics are not comparable without declaring the protocol. The same caution motivates alignment: a same-source multi-protocol cross-check surfaced eight defects that would have shifted the reported numbers across evaluation, training, and statistical caliber, so protocol alignment must precede any continuous-time metric.

The deployment reading stays narrow by design: the system is candidate generation for human triage, emitting suspicious entity pairs for analyst review, not automatic alarms. Within that framing TGN-Sec leads the baselines on pair-level early-warning candidate generation (Table 3); the lead is confined to pair-level recall and F1, while on item-level precision the baselines remain ahead (Table 3). New exogenous-shock pairs should not be predicted, and larger-scale or longer-span re-examination tests whether the modifications and the online loop then show effect.

7. Conclusion

TGN-Sec adapts the temporal graph network with a relation-aware message function and temporal decay attention, and maps link-prediction scores to early-warning candidates through relative ranking; its online maintenance

closes the loop through memory updates, elastic weight consolidation (EWC) fine-tuning, and a Kolmogorov–Smirnov (KS) test that triggers retraining on concept drift.

On early-warning candidate generation TGN-Sec leads TransE, the top baseline, on pair-level F1 (0.1074 ± 0.0134 against TransE at 0.0402) by about 2.2 to 3.0 times, within the candidate-generation-for-human-triage setting. Two results are reported without gloss. The same evaluation also shows lower item-level precision for TGN-Sec (0.0762 ± 0.0090 against baselines of 0.2427 to 0.2813), so no overall advantage is claimed. The two modifications show no significant independent contribution, and recall on entity pairs unseen in training is 0.000 across three seeds, so predictability rests on pairs already observed. The evaluation also shows that aligning training and scoring protocols is necessary before any continuous-time metric is trusted: a same-source and multi-protocol check exposed 8 defects that would have moved the numbers. These findings warrant re-examination on larger-scale or longer-span data.

References

- [1] Zhang, S., Wang, Y., Zhang, Z., et al. (2026). Warning-Graph: An Early Warning Framework for APT Attacks Based on Threat Intelligence Modeling. *IEEE Transactions on Dependable and Secure Computing*.
- [2] Xu, D., Ruan, C., Korpeoglu, E., Kumar, S., & Achan, K. (2020). Inductive representation learning on temporal graphs. In *Proceedings of the International Conference on Learning Representations (ICLR)*.
- [3] Rossi, E., Chamberlain, B., Frasca, F., Eynard, D., Monti, F., & Bronstein, M. (2020). Temporal graph networks for deep learning on dynamic graphs. *arXiv:2006.10637*.
- [4] Nayeri, Z. M., & Rezvani, M. (2026). Alert prediction in computer networks using transformer-based temporal graph neural networks: Identifying the next victim. *Journal of Network and Computer Applications*.
- [5] Dudziak-Gajowiak, D., Juszczyszyn, K., Chudzicki, D. M., et al. (2026). Link prediction using temporal graph neural network model. *Electronics*.
- [6] Trivedi, R., Farajtabar, M., Biswal, P., & Zha, H. (2019). DyRep: Learning representations over dynamic graphs. In *Proceedings of the International Conference on Learning Representations (ICLR)*.
- [7] Alilou, M., Bhuyan, B. P., Fissoune, R., et al. (2026). Temporal motif-based representation learning on continuous-time dynamic graphs. *Data Mining and Knowledge Discovery*.
- [8] Cheriyan, J., Nair, L. S., & Hsiung, P. A. (2026). A unified theoretical and deep learning framework for temporal link prediction in dynamic networks. *IEEE Access*.
- [9] Ren, Y., Xiao, Y., Zhou, Y., et al. (2023). CSKG4APT: A Cybersecurity Knowledge Graph for Advanced Persistent Threat Organization Attribution. *IEEE Transactions on Knowledge and Data Engineering*.
- [10] Ge, W., Wang, J., Cui, Z., et al. (2026). ThreatMAMBA: Achieving high-robustness cyber threat attribution during the evolution of attacks. *IEEE Transactions on Information Forensics and Security*.
- [11] De Paola, A., Drago, S., Ferraro, P., et al. (2026). HOIDS: Concept drift aware hybrid online intrusion detection system. *Journal of Network and Computer Applications*.
- [12] Kirkpatrick, J., Pascanu, R., Rabinowitz, N., Veness, J., Desjardins, G., Rusu, A. A., Milan, K., Quan, J., Ramalho, T., Grabska-Barwinska, A., Hassabis, D., Clopath, C., Kumaran, D., & Hadsell, R. (2017). Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences of the United States of America*, 114(13), 3521–3526.
- [13] Leetaru, K., & Schrod, P. A. (2013). GDELT: Global data on events, location and tone, 1979–2012. In *Proceedings of the International Studies Association (ISA) Annual Convention*.
- [14] Gerner, D. J., Schrod, P. A., & Yilmaz, O. (2002). Conflict and mediation event observations (CAMEO): A new event data framework for the analysis of foreign policy interactions. Presented at the International Studies Association Annual Convention, New Orleans, LA.
- [15] Bordes, A., Usunier, N., Garcia-Duran, A., Weston, J., & Yakhnenko, O. (2013). Translating embeddings for modeling multi-relational data. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2787–2795.
- [16] Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, 9(8), 1735–1780.