iJRASET
International Journal For Research in Applied Science and Engineering Technology



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 14 **Issue:** XI **Month of publication:** November 2026

DOI: https://doi.org/10.22214/ijraset.2026.85579

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue XI Nov 2026- Available at www.ijraset.com

SecureBERT-Fused R-GCN for Cyber Threat Intelligence Knowledge Graph Completion: Extending KGE-SymCL on MITRE ATT&CK

Nallapureddy Pranitha[1], Dr. Ch. D V Subba Rao[2]
[1, 2]*Department of Computer Science and Engineering, S V U College of Engineering, Sri Venkateswara University, Tirupati, India*

Abstract: ***Knowledge graph embedding (KGE) is widely used to complete cyber threat intelligence (CTI) graphs, yet most methods rely on graph structure alone and are validated on general-purpose benchmarks that differ sharply from security data. CTI graphs have a small relation vocabulary, and their entities carry much of their meaning in technical text. This study applies KGE-SymCL, a contrastive learning approach that considers the symmetry of relations, to a cyber threat intelligence (CTI) graph derived from MITRE ATT&CK Enterprise v19.1. The graph comprises 2,595 entities, 20,866 triples, and five distinct relation types. We propose an R-GCN encoder whose structural embeddings are merged with SecureBERT text embeddings through a learned gate. Fifteen KGE backbones are compared, with and without the SymCL term, under an identical protocol over three random seeds. SymCL improves the purely structural R-GCN (MRR 0.0904 to 0.0977) but adds little once SecureBERT fusion is present. The fused model reaches an MRR of 0.1102, a significant gain over plain R-GCN (p = 0.015), although CompGCN and SimKGC remain higher on Hit@1 and Hit@3. These results show that the benefit of contrastive alignment depends on what other signal the encoder already has. To support forensic use, we also add multi-hop path explanations, fusion-gate analysis and tamper-robustness tests that check how predictions behave when the graph is perturbed.***
Keywords: Knowledge Graph Embedding, Cyber Threat Intelligence, MITRE ATT&CK, R-GCN, SecureBERT, Contrastive Learning, Link Prediction, Digital Forensics.
‘

I. INTRODUCTION

Security teams deal with a steady flow of information about attackers, malware, tools and the techniques they use. Cyber threat intelligence (CTI) is the practice of collecting and organizing this information so analysts can make sense of an incident. MITRE ATT&CK provides a systematic classification of adversarial tactics and techniques, while ontologies and knowledge graphs support the integration of related cyber threat intelligence information [1]–[3]. A knowledge graph stores facts as triples, such as a group uses a technique, which makes it easy to query and to reason over [4], [5]. No CTI graph is ever complete, though. Missing links are common, and predicting them, known as knowledge graph completion, can help an investigator trace how an attack unfolded or who was likely behind it.

Knowledge graph embedding (KGE) is the usual approach to this task. These methods map entities and relations into a low-dimensional space and score candidate triples there. Early models treat a relation as a translation [6] or a rotation [7], while others use bilinear or complex-valued scoring [8], [9], or convolutional layers [10]. Graph neural networks go a step further and gather information from an entity's neighbours. R-GCN does this with relation-specific weights [11], and CompGCN adds composition operations between entities and relations [12]. All of these models learn from structure alone.

That limitation matters for CTI. Entities such as malware families, tools and techniques come with long technical descriptions, and a lot of their meaning sits in that text rather than in the links around them. General-purpose language models such as BERT [13] capture text well, but they are not trained on security vocabulary. SecureBERT was built for this gap, since it is pre-trained on cybersecurity text [14]. Text has also been used with contrastive learning for knowledge graph completion, as in SimKGC [15], which depends heavily on the language model it uses.

A different line of work improves KGE through contrastive learning on the graph's own structure. KGE-SymCL builds positive pairs from entities that sit in relation-symmetrical positions, and it aligns them with a loss that needs no negative samples [16]. It works as a plug-in for many KGE models. However, it was evaluated on general benchmarks such as WN18RR, FB15K-237 and NELL-995, which have between 11 and 237 relation types. The MITRE ATT&CK graph we use has only five. It is not clear that symmetrical structure is as useful in a graph this small in relational variety.

©IJRASET: All Rights are Reserved

It is also unclear whether the contrastive term still helps once the model has access to domain text. A knowledge graph was created from MITRE ATT&CK Enterprise v19.1, consisting of 2,595 entities connected through 20,866 triples. KGE-SymCL is then evaluated using fifteen different knowledge graph embedding (KGE) and graph neural network (GNN) backbones. We also propose R-GCN + SecureBERT + Fusion, an encoder that combines structural R-GCN embeddings with SecureBERT embeddings of entity descriptions through a learned gate. Because predictions in an investigation need to be justified, we also add diagnostics for forensic use. The main contributions are:

1) We extend KGE-SymCL from general-purpose benchmarks to a CTI graph with a small relation vocabulary.
2) We propose an R-GCN model fused with SecureBERT through a learned gate, combining structural and textual signals.
3) We run a controlled ablation over the R-GCN family to show how the benefit of SymCL changes when semantic information is added.
4) We add forensic-oriented analysis: multi-hop path explanations, fusion-gate analysis and tamper-robustness tests.
5) We compare all backbones under one protocol, with three random seeds and checkpointed, hash-verified runs.

The subsequent sections of this paper are organized as follows. Section II reviews related work. Section III describes the existing KGE-SymCL method. Section IV presents the proposed SecureBERT-fused R-GCN architecture. Section V explains the experimental setup, Section VI reports the results and analysis, and Section VII concludes the paper with future directions.

II. RELATED WORK

A. *Knowledge Graph Embedding*

KGE models place entities and relations in a continuous space so that true triples score higher than false ones, and surveys cover this area in detail [4], [5]. Translational models treat a relation as a geometric operation. TransE moves the head entity by a relation vector [6], RotatE rotates it in complex space [7], and QuatE and DualE extend rotation to quaternions and dual quaternions to capture richer relation patterns [17], [18]. HAKE uses polar coordinates to reflect hierarchy between entities [19]. Semantic-matching models score a triple through similarity instead. RESCAL factorizes the data with a bilinear tensor [20], DistMult reduces this to a diagonal form [8], ComplEx uses complex embeddings to handle asymmetric relations [9], and ConvE applies convolutions over reshaped embeddings [10]. ComplEx-DURA adds a regularizer to stabilize this factorization [21]. These models are efficient, but each learns from triples only.

B. *Graph Neural Network Encoders*

Graph neural networks incorporate information from neighbouring entities to improve their representations. GCN showed how to pass messages over ordinary graphs [22], but multi-relational graphs need relation-aware designs. R-GCN uses a separate transformation for each relation type [11]. KBGAT adds attention over neighbours and relation features [23], and SCAN builds on structure-aware convolutions [24]. CompGCN combines entity and relation embeddings through composition operations before aggregating [12]. Such encoders usually beat shallow models, but they still depend mainly on the graph's connections.

C. *Contrastive Learning on Knowledge Graphs*

Contrastive learning pulls similar samples together in embedding space, and it has begun to appear in KGE. SimKGC builds its pairs from the similarity of pre-trained language model embeddings [15]. Its quality therefore depends on how well that model understands the domain. KGE-SymCL avoids this by taking positive pairs directly from the graph, using entities in relation-symmetrical positions, and by training with an alignment loss that needs no negatives [16]. In our work, the learned gate used to combine two feature sources follows the idea of gating in highway networks [25].

D. *Language Models for Cybersecurity Text*

General models such as BERT [13] and RoBERTa [26] are trained on everyday text. They are less familiar with malware names, attack techniques and security identifiers. SecureBERT continues RoBERTa's pre-training on cybersecurity text so that these terms are represented better [14]. We use its embeddings as the text signal for entities.

E. *Cyber Threat Intelligence Knowledge Graphs*

MITRE ATT&CK records adversary tactics and techniques observed in real intrusions [1], and it is a common backbone for CTI graphs. Related resources cover defensive countermeasures [3], while ontologies such as UCO aim to standardize threat concepts [2]. These graphs differ from common KGE benchmarks. They use a small and fixed set of relation types, and the meaning of each entity is mostly found in its text description.

F. Explainability, Robustness and Evaluation

In security work, a prediction should come with evidence. DeepPath treats link prediction as a path search, which gives human-readable reasoning chains [27]. Data-poisoning attacks on KGE show that a model can be misled by a few crafted triples [28], which motivates our tamper tests. Because KGE results are normally reported over few random seeds, we follow standard guidance on statistical testing [29] and use ROC analysis for detection tasks [30].

G. Research Gap

KGE-SymCL has been tested mainly on general benchmarks [16]. SecureBERT has shown the value of domain text [14], and R-GCN and CompGCN have shown the value of graph structure [11], [12]. To our knowledge, these three have not been brought together on a MITRE ATT&CK graph. It also remains open whether symmetrical contrastive learning still helps once domain text is already fused into the encoder. This paper tackles that question.

III. BACKGROUND: THE KGE-SYMCL METHOD

This section summarizes the existing framework [16] that our work builds on. The proposed architecture in Section IV is our own contribution and is not part of the original method.

A. Graph and Inverse Graph

A knowledge graph G is a set of triples (e_u, r, e_v), where e_u and e_v are entities from the set E and r is a relation from the set R. To keep track of edge direction, KGE-SymCL builds an inverse graph G^{inv} by flipping every triple, and then works on the union:

$$G_u = G \cup G^{inv} \quad (1)$$

A path between two entities is described by the sequence of relation labels along it. Each label is marked with a plus sign if the edge is followed forward and a minus sign if it is followed backwards.

B. Relation-Symmetrical Structures

Take an anchor entity e_a, a pivot entity e_p and a target entity e_t. The target is symmetrical to the anchor about the pivot when the relation sequence from the anchor to the pivot is the same as the sequence from the target to the pivot, and both have the same length k:

$$RS(e_t, e_p) = RS(e_a, e_p),\ len = k \quad (2)$$

In a CTI graph, a simple example is two techniques that are both used by the same threat group. They sit on either side of the group and share the same one-hop relation to it. The method searches up to a hop limit K and keeps every structure that satisfies Eq. (2). The targets found this way form the candidate positive set for the anchor.

C. Positive Pairs and the Alignment Loss

For each anchor, m targets are sampled at random from its candidate set to keep the cost manageable. The anchor embedding h_a is pulled toward the embedding h_i^+ of each sampled target using a mean squared error on L2-normalized vectors, which reduces to a cosine form:

$$L_{con} = (1/m)\ \Sigma_{i=1}^{m} \left\| h_a/\|h_a\| - h_i^+/\|h_i^+\| \right\|^2 = 2 - (2/m)\ \Sigma_i \cos(h_a, h_i^+) \quad (3)$$

Four properties set this method apart. It is structure-based, because pairs come from the graph rather than from text. It is negative-free, because entities that are not symmetrical are not assumed to be dissimilar. It is augmentation-free, because no altered views of the graph are created. And it does not depend on a language model to build pairs.

D. Training Objective

The contrastive loss is added to the normal task loss with a weight α:

$$L = L_{task} + \alpha \cdot L_{con} \quad (4)$$

The original study reports link prediction on WN18RR, FB15K-237 and NELL-995 and entity classification on AIFB, MUTAG, BGS and AM [16].

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue XI Nov 2026- Available at www.ijraset.com

IV. PROPOSED METHOD: SECUREBERT-FUSED R-GCN

A. Overview

Our architecture uses SecureBERT text in two ways. First, it sets the starting features of an R-GCN, so the structural branch already knows about the text before any message passing begins. Second, the same text vector is sent through a separate semantic branch that skips the graph entirely. A learned gate decides how much of this second branch is added back to the graph output. This differs from a plain R-GCN, which starts from freely learned embeddings, and from a simple two-branch model whose graph branch ignores text. Fig. 1 shows the overall design.

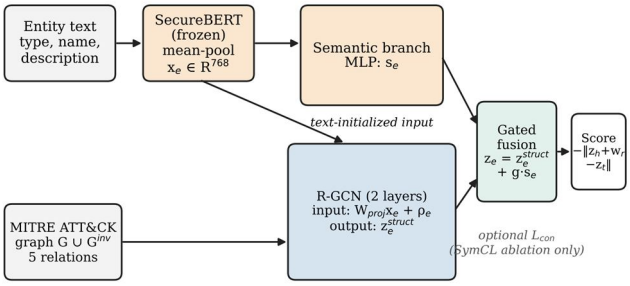


Fig. 1. Architecture of the proposed SecureBERT-fused R-GCN. The dashed SymCL term is used only in the ablation.

B. Text Representation with SecureBERT

Every entity gets a short description built from its type, name and description field, in the form "type: ⟨type⟩. name: ⟨name⟩. description: ⟨description⟩". If the name or description is missing, the type and identifier are used instead. The text, truncated to 128 tokens, is encoded by SecureBERT [14]. The model is frozen, so it is only a feature extractor. Each embedding is the attention-masked mean of the last hidden layer:

$$x_e = MeanPool(SecureBERT(text_e)) \in R^{768} \quad (5)$$

All 2,595 vectors are computed once and cached, so no language model runs during KGE training.

C. SecureBERT-Initialized Structural Branch

The structural branch is an R-GCN with basis decomposition [11] that runs over the union graph G_u. Its input for each entity combines a projected SecureBERT vector with a learnable residual embedding, which leaves room for information that the text does not hold:

$$z_e^{(0)} = W_{proj}\, x_e + \rho_e \quad (6)$$

Two R-GCN layers then spread this signal across the graph:

$$z_e^{(l+1)} = \sigma\left(\Sigma_r \Sigma_{u \in Nr(e)}\, c_{e,r}\, W_r\, z_u^{(l)} + W_0\, z_e^{(l)} \right) \quad (7)$$

$$W_r = \Sigma_{b=1}^{B}\, a_{r,b}\, V_b \quad (8)$$

Here $N_r(e)$ is the set of neighbours of e under relation r, $c_{e,r}$ is a square-root degree normalization, W_0 is a self-loop weight, and the relation weights are built from B = 6 shared basis matrices. The output of the last layer, z_e^{struct}, is therefore a text-aware, graph-contextualized representation rather than a purely structural one.

D. Independent Semantic Branch and Gated Fusion

In parallel, a small MLP maps the same SecureBERT vector to the model dimension without any message passing:

$$s_e = W_2\, ReLU(W_1\, x_e + b_1) + b_2 \quad (9)$$

The two branches are then combined through a sigmoid gate with a single learnable scalar γ, initialized at −1 so that the gate starts near 0.27:

$$z_e = z_e^{struct} + g \cdot s_e,\ g = \sigma(\gamma) \quad (10)$$

The gate lets the model learn from data how strongly to bring back the raw text signal on top of what the graph has already produced. All headline results use this single global gate.

©IJRASET: All Rights are Reserved

E. Scoring Function and Training Loss

Triples are scored with a TransE-style distance on the fused embeddings, where w_r is a learned relation vector:

$$score(h, r, t) = -|| z_h + w_r - z_t ||_2 \quad (11)$$

Training uses a margin ranking loss against one negative per positive triple. The negative is made by replacing the head or the tail, chosen at random, with an entity that does not form a true triple anywhere in the dataset:

$$L_{task} = (1/|B|) \Sigma ReLU(1 - score(h, r, t) + score(h', r, t')) \quad (12)$$

F. Optional SymCL Term (Ablation Only)

The alignment loss from Section III can be added on top of this model with K = 2, 64 sampled anchors per step, six positives per anchor and $\alpha = 0.001$, giving $L = L_{task} + \alpha \cdot L_{con}$. We treat this strictly as an ablation, not as part of the proposed model, because our results show it does not help once fusion is present. Since the MITRE graph has very dense neighbourhoods around the uses relation, symmetry extraction needed three practical caps: a local degree limit of 200 when expanding a hop, a limit of 200 members per symmetry group, and subsampling of the source triples when they exceed a set size. All SymCL results use this capped extraction.

We also tested three exploratory variants on the same base: a domain-aware version that builds symmetry groups separately for offensive (uses), defensive (mitigates, detects), hierarchical (subtechnique-of) and attribution (attributed-to) relations; a density-aware version that reweights anchor sampling by $1/\sqrt{(1+\deg(e))}$; and a type-conditioned gate with one value per entity type.

G. Forensic Diagnostics

Because the encoder works over explicit and readable relations, predicted links can be traced to multi-hop paths in the training graph. We also inspect the gate value and the norms of the two branches, and test how well prediction margins separate genuine triples from tampered ones. These are reported in Section VI.

V. EXPERIMENTAL SETUP

A. Dataset

The dataset was obtained from the MITRE ATT&CK Enterprise v19.1 release, provided in STIX 2.1 format [1]. Revoked and deprecated objects were removed, and five relation types were kept. The file's SHA-256 checksum was verified before use. Table I gives the statistics. The graph is dominated by the uses relation, which accounts for about 87% of all triples. Edges were split per relation, 80/10/10, with a fixed seed so that every relation appears in all splits.

Because the uses relation accounts for most triples (18,220 of 20,866), a model can score well largely by learning this one relation. The other four relations are much smaller: mitigates (1,448), detects (697), subtechnique-of (475) and attributed-to (26). With stratified splitting, attributed-to leaves only a handful of test triples, so any per-relation result for it should be read with caution.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

TABLE I
STATISTICS OF THE MITRE ATT&CK GRAPH

Item	Value
Source / version	attack-stix-data, enterprise-attack-19.1
Entities	2,595
Triples (total)	20,866
Train / Valid / Test	16,693 / 2,088 / 2,085
Relation types	5
Relation counts	uses 18,220; mitigates 1,448; detects 697; subtechnique-of 475; attributed-to 26
Split	Stratified per relation, 80/10/10, seed 42

B. Baselines

Fifteen backbones are compared: TransE [6], RotatE [7], QuatE [17], DualE [18], HAKE [19], RESCAL [20], DistMult [8], ComplEx [9], ConvE [10], ComplEx-DURA [21], R-GCN [11], SCAN [24], KBGAT [23], CompGCN [12] and SimKGC [15]. Each one is trained twice, with and without the SymCL term.

C. Implementation Details

All models use the settings in Table II and the same three seeds (42, 7 and 123). There is no early stopping or checkpoint selection: every model trains for the full 30 epochs and the final weights are evaluated. For the proposed model, the semantic branch and gate stay frozen during the first two epochs so the structural branch can adapt first, and the gate is trained at twice the base learning rate. Weights of every run are stored with a SHA-256 checksum.

TABLE II
TRAINING HYPERPARAMETERS

Setting	Value
Embedding dimension	64
R-GCN layers / bases	2 / 6
Normalization	Square-root degree
Batch size / negatives per positive	256 / 1
Optimizer	Adam, weight decay 1×10^{-5}
Learning rate (base / gate)	0.005 / 0.01
Epochs (warm-up)	30 (2)
SymCL: K, m, positives per anchor, α	2, 64, 6, 0.001

D. Evaluation

For every test triple, the correct entity is ranked against all candidate entities, excluding other previously known valid triples. Performance is measured using mean reciprocal rank (MRR), Hit@1, Hit@3, Hit@10, and mean rank (MR). Numbers are averaged over the three seeds. Differences are checked with paired t-tests across seeds. With only three seeds these tests have low power, so we report p-values as supporting evidence and not as proof.

TABLE III(A)
LINK PREDICTION WITHOUT SYMCL (MEAN OF 3 SEEDS)

Method	MRR	Hit@1	Hit@3	Hit@10	MR
TransE	.1061	.0443	.1039	.2275	171.3
RotatE	.0728	.0301	.0694	.1500	319.8
QuatE	.1099	.0568	.1129	.2135	314.2
DualE	.0965	.0466	.0966	.1901	322.6
HAKE	.0442	.0122	.0336	.0957	287.8
RESCAL	.0806	.0334	.0793	.1680	305.1
DistMult	.0990	.0486	.0999	.1932	331.7
ComplEx	.0995	.0492	.1008	.1926	324.7
ConvE	.0973	.0455	.0976	.1969	374.8
ComplEx-DURA	.0908	.0349	.0852	.1978	199.5
R-GCN	.0904	.0376	.0858	.1919	184.5
SCAN	.1016	.0476	.1004	.2070	216.2
KBGAT	.0857	.0332	.0816	.1825	187.8
CompGCN	.1126	.0536	.1155	.2240	168.2
SimKGC	.1094	.0608	.1163	.2013	375.7

TABLE III(B)
LINK PREDICTION WITH SYMCL (MEAN OF 3 SEEDS)

Method	MRR	Hit@1	Hit@3	Hit@10	MR
TransE	.1096	.0475	.1084	.2325	169.0
RotatE	.0688	.0276	.0635	.1420	321.0
QuatE	.1076	.0543	.1109	.2076	305.2
DualE	.0970	.0470	.0992	.1902	321.4
HAKE	.0454	.0131	.0363	.0970	282.8
RESCAL	.0815	.0373	.0779	.1640	313.4
DistMult	.0984	.0488	.0986	.1914	326.5
ComplEx	.1006	.0517	.1006	.1934	326.5
ConvE	.0972	.0466	.0962	.1941	376.7
ComplEx-DURA	.0890	.0338	.0837	.1940	203.2
R-GCN	.0977	.0428	.0946	.2045	187.0
SCAN	.1016	.0480	.0998	.2069	213.7
KBGAT	.0837	.0324	.0755	.1836	185.7
CompGCN	.1073	.0476	.1082	.2210	171.8
SimKGC	.1088	.0621	.1156	.1941	365.1

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

VI. RESULTS AND DISCUSSION

A. Fifteen Backbones with and Without SymCL

Tables III(a) and III(b) compare the backbones. CompGCN has the best baseline MRR (0.1126) and mean rank (168.2), and SimKGC has the best baseline Hit@1 (0.0608) and Hit@3 (0.1163). With SymCL, SimKGC reaches the highest Hit@1 of all rows (0.0621) and TransE the highest Hit@10 (0.2325). Adding SymCL raises MRR for six backbones (TransE, DualE, HAKE, RESCAL, ComplEx and R-GCN), leaves SCAN unchanged and lowers it for the other eight. The largest drops are for CompGCN (−0.0053) and RotatE (−0.0040). SymCL is therefore not a uniform gain. Its value depends on how much structural signal the base encoder already extracts (Tables III(a), III(b) and Fig. 2). For reference, the proposed fused R-GCN (Table IV) obtains Hit@1 of 0.0510 and Hit@3 of 0.1102, which is below CompGCN (0.0536 and 0.1155) and SimKGC on these two metrics. Its advantage is therefore over the plain R-GCN and, in mean rank (154.8), over every baseline in Tables III(a) and III(b), not in top-rank accuracy.

B. R-GCN and CompGCN Ablations

Table IV and Fig. 3 isolate the effect of SecureBERT, fusion and SymCL. For R-GCN, three patterns appear. First, SecureBERT initialization alone raises MRR from 0.0904 to 0.0989 and lowers mean rank from 184.5 to 166.6. Second, gated fusion adds more, reaching an MRR of 0.1102 (21.9% above plain R-GCN) and the best mean rank of the family, 154.8. Third, the effect of SymCL reverses once fusion is present. On plain R-GCN it helps (0.0904 to 0.0977), but on the fused model it lowers MRR (0.1102 to 0.1071), Hit@1 (0.0510 to 0.0478), Hit@3 (0.1102 to 0.1083) and Hit@10 (0.2228 to 0.2212), and it worsens mean rank (154.8 to 160.3). The same reversal appears for CompGCN (fused model, 0.1135 to 0.1085). For CompGCN, SecureBERT improves mean rank (168.2 to 155.9) but leaves MRR essentially unchanged.

C. Statistical Significance

Table V reports paired t-tests on MRR. The fused R-GCN is significantly better than plain R-GCN ($p = 0.0154$) and better than R-GCN with SymCL ($p = 0.0147$). SecureBERT initialization alone also gives a significant gain ($p = 0.0330$). In contrast, adding SymCL to the fused model changes MRR by −0.0031, which cannot be told apart from zero ($p = 0.584$). SymCL alone over plain R-GCN is positive but not significant ($p = 0.133$). With three seeds the tests are underpowered, so the negative effect of SymCL on the fused model should be read as a direction and not as a proven result.

TABLE IV
ABLATION FAMILIES (MEAN OF 3 SEEDS)

Configuration	MRR	Hit@1	Hit@3	Hit@10	MR
R-GCN	0.0904	0.0376	0.0858	0.1919	184.5
R-GCN + SymCL	0.0977	0.0428	0.0946	0.2045	187.0
R-GCN + SecureBERT	0.0989	0.0419	0.0974	0.2086	166.6
R-GCN + SecureBERT + SymCL	0.1030	0.0438	0.1018	0.2165	167.1
R-GCN + SecureBERT + Fusion (proposed)	0.1102	0.0510	0.1102	0.2228	154.8
R-GCN + SecureBERT + Fusion + SymCL	0.1071	0.0478	0.1083	0.2212	160.3
CompGCN	0.1126	0.0536	0.1155	0.2240	168.2
CompGCN + SymCL	0.1073	0.0476	0.1082	0.2210	171.8
CompGCN + SecureBERT	0.1118	0.0513	0.1135	0.2271	155.9
CompGCN + SecureBERT + SymCL	0.1131	0.0522	0.1128	0.2329	151.5
CompGCN + SecureBERT + Fusion	0.1135	0.0537	0.1120	0.2300	151.3
CompGCN + SecureBERT + Fusion + SymCL	0.1085	0.0480	0.1062	0.2284	154.0

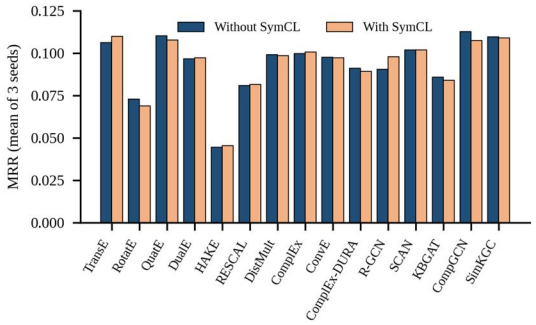


Fig. 2. MRR of the fifteen backbones with and without SymCL (mean of 3 seeds).

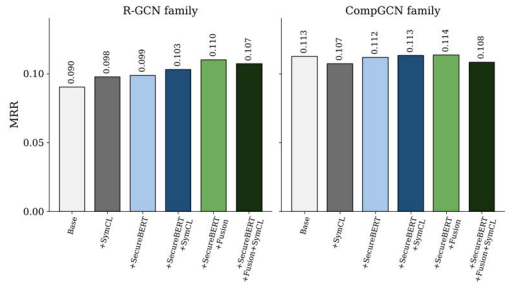


Fig. 3. MRR for the R-GCN and CompGCN ablation families. Bars show the mean of three seeds; the darker bar in each color pair includes SymCL.

TABLE V
PAIRED T-TESTS ON MRR (N = 3 SEEDS)

Comparison	**Mean ΔMRR**	**t**	**p**
Fused R-GCN vs. R-GCN	+0.0198	7.97	0.0154
R-GCN + SecureBERT vs. R-GCN	+0.0085	5.36	0.0330
Fused R-GCN vs. R-GCN + SecureBERT	+0.0113	3.83	0.0619
R-GCN + SymCL vs. R-GCN	+0.0073	2.47	0.1325
Fused R-GCN vs. R-GCN + SymCL	+0.0125	8.15	0.0147
Fused R-GCN + SymCL vs. fused R-GCN	−0.0031	−0.65	0.5842
CompGCN + SymCL vs. CompGCN	−0.0053	−1.11	0.3841
Fused R-GCN vs. fused CompGCN	−0.0033	−1.11	0.3816

The reason R-GCN is our host encoder is not that it has the highest absolute score. The fused CompGCN has a slightly higher mean MRR (0.1135 against 0.1102), while the fused R-GCN has the higher best-seed MRR (0.1162 against 0.1148). SecureBERT and fusion produce significant improvements when applied to R-GCN. However, CompGCN, which performs better as a baseline without text features, shows no statistically significant change in MRR after incorporating SecureBERT ($p = 0.872$) or fusion ($p = 0.421$).

D. Sensitivity of the SymCL Settings

We swept the contrastive weight α over {0, 0.0001, 0.001, 0.01} on TransE with SymCL, using 8-epoch runs. Mean MRR moved only between 0.0739 and 0.0767, which is small compared with the seed-to-seed spread, so α = 0.001 is a controlled choice and not the source of our findings. We also varied the group-size cap (20, 50, 100, 200) and the maximum-degree limit (10, 50, 100, 200, none). Neither sweep showed a significant trend (cap 200 against 20: p = 0.72; no limit against 10: p = 0.55). The three exploratory variants of Section IV-F did not help either. Against the fused SymCL baseline (MRR 0.1071), the domain-aware version reached 0.1027, the density-aware version 0.1059 and the type-conditioned gate 0.0962.

E. Fusion-Gate Analysis

For the proposed model (seed 42), the gate value is 0.3150. Across all entities, the mean norm of the structural representation is 5.31 and that of the transformed semantic branch is 9.52, which falls to 3.00 after gating. The semantic share of the final vector has a mean of 0.379 and ranges from 0.150 to 0.740 across entities, so the text contributes noticeably but unevenly. The gate should be read as the weight of the independent semantic branch, not as the total share of text, since text is already present inside the structural branch. In the type-conditioned variant, gate values differ by entity type; for example, data components reach 0.7065 while detection strategies are lowest at 0.1799.

F. Path-Based Explainability

We checked whether test triples have a witness path in which head and tail share a pivot under a relation-symmetrical structure of at most two hops. This holds for 13.0% of sampled test triples. Coverage depends strongly on the relation: 33.3% for subtechnique-of, 13.7% for uses, and none for detects and mitigates at this hop limit. This is a measure of how many predictions carry this specific kind of evidence, not a claim that all predictions can be explained this way.

G. Forensic Evidence Reliability

We tested how well scores separate genuine triples from three kinds of perturbation (Table VI and Fig. 4). The margin-based score separates them well, with AUROC between 0.92 and 0.94. Symmetry co-membership alone is close to chance (0.49 to 0.50), so it carries almost no forensic signal on its own, and the combined detector falls in between. We therefore keep the margin score as the main forensic indicator.

TABLE VI
PERTURBATION DETECTION (AUROC)

Test	Margin only	Symmetry only	Combined
False-edge substitution	0.936	0.486	0.867
Relation corruption	0.920	0.500	0.881
False-alternative injection	0.927	0.495	0.868

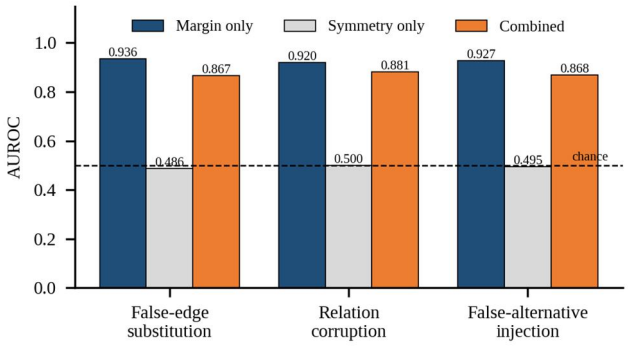


Fig. 4. AUROC comparison of the three detectors across different perturbation types. The dashed horizontal line indicates the chance-level performance of 0.5.

H. Computational Cost

SecureBERT embeddings are stored in advance and reused during training, eliminating the need for repeated language-model inference. The extra cost comes only from the semantic MLP, the gate and, optionally, SymCL. In two short single-seed timing runs of five epochs, the proposed model took 60.58 s against 52.78 s for plain R-GCN (+14.8%), and the full pipeline with SymCL took 63.32 s against 55.06 s (+15.0%). Because wall-clock times vary between runs, the overhead is best read as roughly 15%.

I. Supplementary Entity Classification and Reproducibility

For comparison with the source study, we also ran R-GCN, CompGCN and RDF2Vec with and without SymCL on AIFB, MUTAG, BGS and AM. The effect of SymCL was again dataset-dependent. For instance, it helped R-GCN on MUTAG (65.69% to 66.67%) and BGS (83.91% to 85.06%) and left AIFB unchanged, while CompGCN was identical with and without SymCL on all four datasets. Some of our reproduced baselines are well below the values in the source study, and CompGCN on AM (34.85%) looks like a collapsed prediction, so we report these results only as supplementary. All 114 link-prediction checkpoints (38 models × 3 seeds) and the cached SecureBERT table were hash-verified, so each reported number traces to a stored run.

VII. CONCLUSION AND FUTURE SCOPE

We investigated how well KGE-SymCL performs when working with cyber threat data represented as a graph derived from MITRE ATT&CK, rather than using only standard benchmark datasets. We proposed an R-GCN that combines graph structure with SecureBERT text through a learned gate. It reached a mean MRR of 0.1102 compared with 0.0904 for plain R-GCN, a significant gain ($p = 0.015$). Relation-symmetrical contrastive learning helped the structure-only R-GCN, but it gave no clear extra benefit once the text signal was fused in. In short, its value depends on what else the encoder already knows. The study also showed that the CompGCN variants reach comparable results, so the proposed model is a domain-focused design and not a universally superior one. The forensic checks showed that margin-based scores detect tampering far better than symmetry-based signals.

Future work can follow five directions: building reliable negative pairs for contrastive learning, since non-symmetrical entities are not guaranteed to be true negatives; learning relation groups automatically instead of defining them by hand; fine-tuning SecureBERT jointly with the graph encoder; testing the fusion on other security graphs such as CVE, CWE or D3FEND; and turning witness paths into natural-language explanations for analysts.

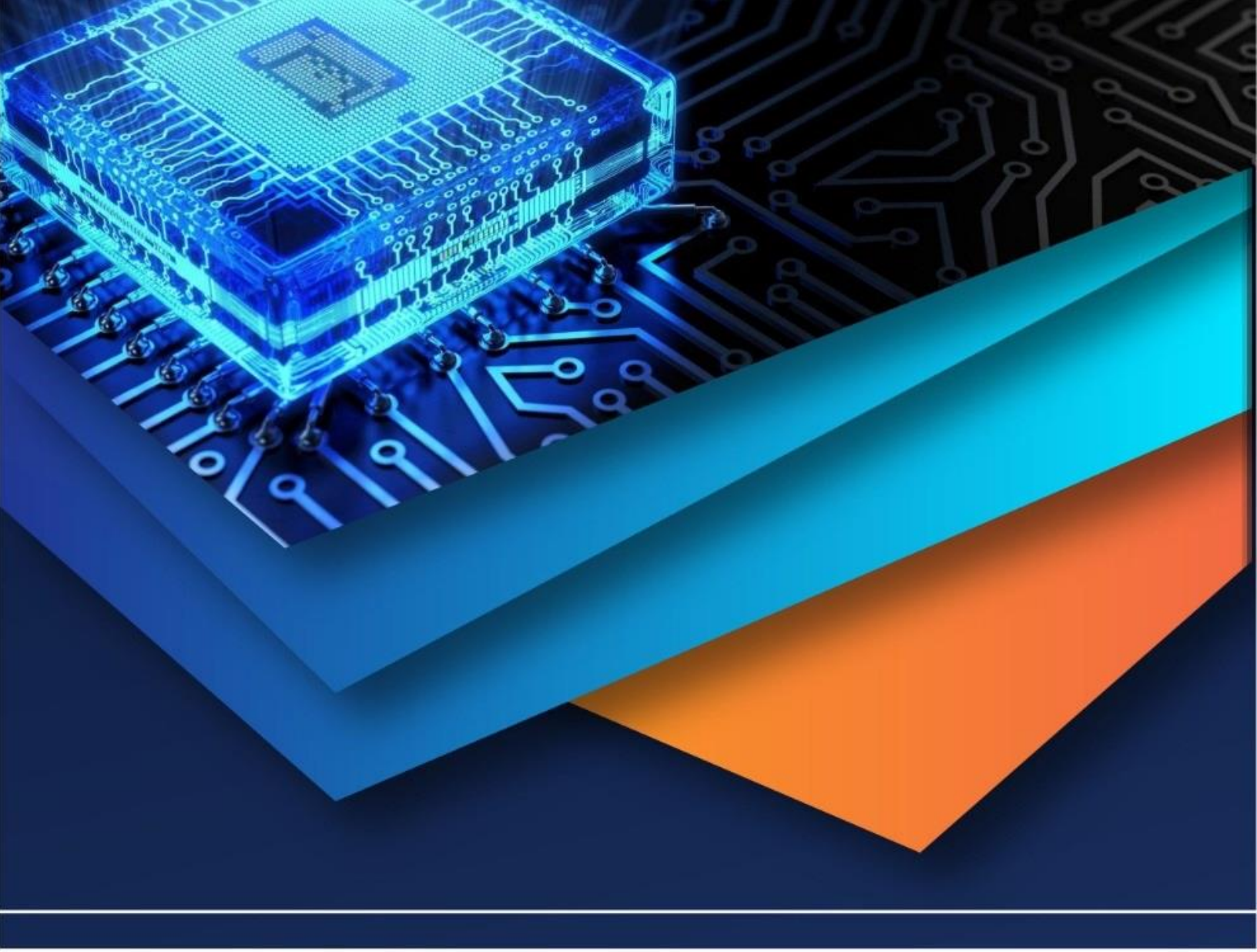
REFERENCES

[1] B. E. Strom, A. Applebaum, D. P. Miller, K. C. Nickels, A. G. Pennington, and C. B. Thomas, "MITRE ATT&CK: Design and philosophy," MITRE Corporation, Tech. Rep., 2020.
[2] Z. Syed, A. Padia, T. Finin, L. Mathews, and A. Joshi, "UCO: A unified cybersecurity ontology," in Proc. AAAI Workshop on Artificial Intelligence for Cyber Security, 2016.
[3] P. Kaloroumakis and M. Robinson, "Toward a knowledge graph of cybersecurity countermeasures," MITRE Corporation, Tech. Rep., 2021.
[4] Q. Wang, Z. Mao, B. Wang, and L. Guo, "Knowledge graph embedding: A survey of approaches and applications," IEEE Trans. Knowl. Data Eng., vol. 29, no. 12, pp. 2724–2743, 2017.
[5] S. Ji, S. Pan, E. Cambria, P. Marttinen, and P. S. Yu, "A survey on knowledge graphs: Representation, acquisition, and applications," IEEE Trans. Neural Netw. Learn. Syst., vol. 33, no. 2, pp. 494–514, 2021.
[6] A. Bordes, N. Usunier, A. Garcia-Duran, J. Weston, and O. Yakhnenko, "Translating embeddings for modeling multi-relational data," in Adv. Neural Inf. Process. Syst., vol. 26, 2013.
[7] Z. Sun, Z.-H. Deng, J.-Y. Nie, and J. Tang, "RotatE: Knowledge graph embedding by relational rotation in complex space," in Proc. ICLR, 2019.
[8] B. Yang, W.-t. Yih, X. He, J. Gao, and L. Deng, "Embedding entities and relations for learning and inference in knowledge bases," in Proc. ICLR, 2015.
[9] T. Trouillon, J. Welbl, S. Riedel, É. Gaussier, and G. Bouchard, "Complex embeddings for simple link prediction," in Proc. ICML, 2016, pp. 2071–2080.
[10] T. Dettmers, P. Minervini, P. Stenetorp, and S. Riedel, "Convolutional 2D knowledge graph embeddings," in Proc. AAAI, 2018.
[11] M. Schlichtkrull, T. N. Kipf, P. Bloem, R. van den Berg, I. Titov, and M. Welling, "Modeling relational data with graph convolutional networks," in Proc. ESWC, 2018, pp. 593–607.
[12] S. Vashishth, S. Sanyal, V. Nitin, and P. Talukdar, "Composition-based multi-relational graph convolutional networks," in Proc. ICLR, 2019.
[13] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in Proc. NAACL-HLT, 2019, pp. 4171–4186.
[14] E. Aghaei, X. Niu, W. Shadid, and E. Al-Shaer, "SecureBERT: A domain-specific language model for cybersecurity," in Proc. SecureComm, 2022, pp. 39–56.
[15] L. Wang, W. Zhao, Z. Wei, and J. Liu, "SimKGC: Simple contrastive knowledge graph completion with pre-trained language models," in Proc. ACL, 2022, pp. 4281–4294.
[16] K. Liang, Y. Liu, S. Zhou, W. Tu, Y. Wen, X. Yang, X. Dong, and X. Liu, "Knowledge graph contrastive learning based on relation-symmetrical structure," IEEE Trans. Knowl. Data Eng., 2023.
[17] S. Zhang, Y. Tay, L. Yao, and Q. Liu, "Quaternion knowledge graph embeddings," in Adv. Neural Inf. Process. Syst., vol. 32, 2019.
[18] Z. Cao, Q. Xu, Z. Yang, X. Cao, and Q. Huang, "Dual quaternion knowledge graph embeddings," in Proc. AAAI, 2021, pp. 6894–6902.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

[19] Z. Zhang, J. Cai, Y. Zhang, and J. Wang, "Learning hierarchy-aware knowledge graph embeddings for link prediction," in Proc. AAAI, 2020, pp. 3065–3072.
[20] M. Nickel, V. Tresp, and H.-P. Kriegel, "A three-way model for collective learning on multi-relational data," in Proc. ICML, 2011.
[21] Z. Zhang, J. Cai, and J. Wang, "Duality-induced regularizer for tensor factorization based knowledge graph completion," in Adv. Neural Inf. Process. Syst., vol. 33, 2020, pp. 21604–21615.
[22] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in Proc. ICLR, 2017.
[23] D. Nathani, J. Chauhan, C. Sharma, and M. Kaul, "Learning attention-based embeddings for relation prediction in knowledge graphs," in Proc. ACL, 2019.
[24] C. Shang, Y. Tang, J. Huang, J. Bi, X. He, and B. Zhou, "End-to-end structure-aware convolutional networks for knowledge base completion," in Proc. AAAI, 2019, pp. 3060–3067.
[25] R. K. Srivastava, K. Greff, and J. Schmidhuber, "Highway networks," arXiv:1505.00387, 2015.
[26] Y. Liu et al., "RoBERTa: A robustly optimized BERT pretraining approach," arXiv:1907.11692, 2019.
[27] W. Xiong, T. Hoang, and W. Y. Wang, "DeepPath: A reinforcement learning method for knowledge graph reasoning," in Proc. EMNLP, 2017, pp. 564–573.
[28] H. Zhang et al., "Data poisoning attack against knowledge graph embedding," in Proc. IJCAI, 2019, pp. 4853–4859.
[29] T. G. Dietterich, "Approximate statistical tests for comparing supervised classification learning algorithms," Neural Comput., vol. 10, no. 7, pp. 1895–1923, 1998.
[30] T. Fawcett, "An introduction to ROC analysis," Pattern Recognit. Lett., vol. 27, no. 8, pp. 861–874, 2006.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)