mathematics

MDPI

Article

Research on Spurious-Negative Sample Augmentation-Based Quality Evaluation Method for Cybersecurity Knowledge Graph

Bin Chen [1], Hongyi Li [1] and Ze Shi [1,2,*]

[1] School of Cyber Science and Technology, Beihang University, Beijing 100191, China
[2] School of Artificial Intelligence, Chongqing Technology and Business University, Chongqing 400067, China
* Correspondence: zb2039107@buaa.edu.cn

Abstract: As the forms of cyber threats become increasingly severe, cybersecurity knowledge graphs (KGs) have become essential tools for understanding and mitigating these threats. However, the quality of the KG is critical to its effectiveness in cybersecurity applications. In this paper, we propose a spurious-negative sample augmentation-based quality evaluation method for cybersecurity KGs (SNAQE) that includes two key modules: the multi-scale spurious-negative triple detection module and the adaptive mixup based on the attention mechanism module. The multi-scale spurious-negative triple detection module classifies the sampled negative triples into spurious-negative and true-negative triples. Subsequently, the attention mechanism-based adaptive mixup module selects appropriate mixup targets for each spurious-negative triple, constructing partially correct triples and achieving more precise sample generation in the entity embedding space to assist in training the KG quality evaluation models. Through extensive experimental validation, the SNAQE model not only performs excellently in general-domain KG quality evaluation but also achieves outstanding outcomes in the cybersecurity KGs, significantly enhancing the accuracy and F1 score of the model, with the best F1 score of 0.969 achieved on the FB15K dataset.

Keywords: cybersecurity; quality evaluation; spurious-negative triple; knowledge graphs

MSC: 68T50



Academic Editor: Tianyong Hao

Received: 17 November 2024
Revised: 25 December 2024
Accepted: 26 December 2024
Published: 27 December 2024

Citation: Chen, B.; Li, H.; Shi, Z. Research on Spurious-Negative Sample Augmentation-Based Quality Evaluation Method for Cybersecurity Knowledge Graph. *Mathematics* **2025**, *13*, 68. https://doi.org/10.3390/math13010068

Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

With the increasing complexity of cyberattack methods and the rapid growth of data scale, cybersecurity KGs have played an irreplaceable role in threat intelligence analysis, intrusion detection, and situational awareness, providing new solutions for the comprehensive integration of dispersed cybersecurity data and enhancing threat response capabilities [1–3]. However, since the construction of KGs often relies on automated or semi-automated techniques, it inevitably introduces noise and erroneous information. The performance of downstream applications can be significantly compromised by a low-quality KG, resulting in issues such as false positives, undetected threats, and flawed deductions and decision-making. Hence, accurately evaluating the quality of KGs to guarantee the dependability and correctness of their triple data is an important research challenge that needs to be addressed [4,5].

Current knowledge representation learning models have demonstrated notable success in tasks such as triple classification and KG completion by mapping entities and relations to low-dimensional vector spaces [6–8]. During the model training process, negative sampling plays a crucial role. If the quality of negative samples is too low, leading to a large

discrepancy between positive and negative samples, it can result in the gradient vanishing problem. Negative sampling techniques aim to generate high-quality negative samples to increase the score difference between positive and negative samples, thereby effectively enhancing the training performance of the model.

However, traditional negative sampling methods are generally based on the closed-world assumption, which posits that all non-existent triples are negative samples [6]. These negative sampling methods largely overlook the issue of sampling noisy triples, often treating high-scoring but non-existent triples as high-quality negative triples. However, such high-scoring negative triples may actually be true triples, i.e., triples that do not exist in the knowledge graph but are factually correct. This may lead to the introduction of spurious-negative samples. We define spurious-negative triples as those absent from the knowledge graph but factually correct, using "score" as the criterion to determine whether a triple is a spurious-negative triple. Spurious-negative samples provide incorrect supervisory signals to the model, thereby misleading the training process and reducing its prediction accuracy. Therefore, in the task of quality evaluation for cybersecurity knowledge graphs, it is particularly important to mitigate the impact of spurious-negative samples during the sampling process in model training.

In this paper, we propose a spurious-negative sample augmentation-based quality evaluation method for cybersecurity KGs (SNAQE). SNAQE divides the sampled negative triples into critical spurious-negative triples and true-negative triples through a multi-scale spurious-negative triple detection module. Subsequently, the adaptive mixup based on the attention mechanism module selects suitable mixup targets for each spurious-negative triple. This adaptive mixup strategy can automatically adjust the mixing ratio based on the features between samples, thereby generating more reasonable training samples and enhancing the performance of the model and generalization ability. Through extensive comparative experiments on both general-domain datasets and cybersecurity datasets, we demonstrate the superiority of our KG quality evaluation model. To summarize, the key contributions of this study are as follows:

(1) Key Technical Contributions

- We designed a multi-scale spurious-negative triple detection method, which improves the model's ability to distinguish spurious-negative samples by performing multi-scale scoring on negatively sampled triples.
- We proposed an adaptive mixup based on the attention mechanism, which generates more reasonable training samples through adaptive mixing between samples, thereby enhancing the model's accuracy and F1 score in the knowledge graph quality evaluation task.

(2) Additional Contributions

- Through extensive experiments, we demonstrated the effectiveness of our model and its key components in the knowledge graph quality evaluation task, achieving the current State-of-the-Art results.

The remainder of the paper is structured as follows: Section 2 reviews existing research on KG quality assessment and explores methods for negative sampling in KGs. Section 3 offers a comprehensive overview of the proposed model. Section 4 presents the experimental outcomes and provides an in-depth analysis and discussion of the results. Finally, Section 5 concludes the paper and suggests potential avenues for future work.

2. Related Work

With the widespread application of knowledge graphs in fields such as semantic search, intelligent recommendation, and knowledge reasoning, the importance of knowledge graph

quality evaluation has become increasingly prominent [9–11]. However, as the scale of knowledge graphs continues to expand, effectively evaluating their quality has emerged as a critical challenge. Against this backdrop, negative sampling techniques play a crucial role in the quality evaluation of knowledge graphs. This section reviews related work in the fields of knowledge graph quality evaluation and negative sampling methods.

2.1. KG Quality Evaluation

Conventional approaches to KG quality evaluation typically rely on manual inspection and rule formulation, which have limitations when dealing with large-scale data and complex relationships. In recent years, researchers have started investigating the application of machine-learning and deep-learning techniques to automate KG quality evaluation, with the goal of enhancing the effectiveness and accuracy of evaluations, thereby better accommodating the complexities and scale of large datasets. Xie et al. [12] introduced CKRL, a KG quality evaluation model based on translational distance that effectively utilizes the global structural features for error detection. Through comprehensive experiments, the capability of the model to assess the confidence of triples by leveraging structural details in KG quality evaluation tasks was validated. Inspired by the successful application of automated machine learning (AutoML) technology, Zhang et al. [13] proposed AutoSF, an approach that automatically designs scoring functions suitable for different KGs through AutoML. This method constructs a search space by defining a unified representation for scoring functions and proposes a greedy algorithm to optimize the search process. Experimental results show that AutoSF outperforms manually designed scoring functions in evaluating KG quality, highlighting the benefits of automating the design of scoring functions. Jia et al. [14] introduced a model for measuring the confidence of triples, utilizing a cross neural network architecture that integrates the internal semantic properties of triples with global reasoning context. The model achieves confidence measurement and fusion at the entity, relation, and global layers. Experimental results indicate that the model achieved good evaluation accuracy in KG error detection tasks. Bougiatiotis et al. [15] introduced the PRGE method, which combines path ranking scores of triples with graph embedding frameworks, making the generated embeddings more robust to noise in KGs and improving KG quality evaluation performance across multiple datasets.

Table 1 provides a summary of the accuracy and F1 scores of different models in assessing the quality of KGs using the FB15K dataset. Among these, the DistMult model [13,16] cleverly applies the bilinear function $h^T RT$ to deeply explore the intrinsic relationships between entities and relations, achieving a quality evaluation accuracy of 0.808. The ComplEx model [13,17], which uses complex-valued vectors to represent entities and relations, enhances the model's accuracy in handling asymmetric relations and achieves an accuracy of 0.818 on the FB15K dataset. However, when faced with challenges in modeling complex relationships such as one-to-many, many-to-one, and many-to-many, the TransE model exhibits certain limitations. It struggles to fully capture the diversity and complexity of relations, leading to poor performance in quality evaluation for complex graphs. Subsequently, researchers introduced relation-specific parameters and developed models like TransD, TransR, and TransH, which led to notable improvements in the performance of evaluating KG quality [14,18,19]. Based on the Tucker model [20], Li et al. introduced the AttTucker model, which incorporates multi-head attention mechanisms and models the KG using a central tensor along with three-factor matrices, resulting in notable enhancements on the FB15K dataset.

Table 1. Performance comparison of various models on the FB15K dataset.

Models	Accuracy	F1 Score
NTN [18]	0.665	-
SimplE [13]	0.815	
DistMult [13]	0.808	-
CKRL(LT + PP + AP) [12]	0.817	-
CKRL(LT + PP) [12]	0.819	-
CKRL(LT) [12]	0.818	-
ComplEx [13]	0.818	-
AutoSF [13]	0.827	-
Analogy [13]	0.821	-
TransT (TT) [21]	0.822	-
TransT (TT + DT) [21]	0.824	-
TransE [14]	0.868	0.876
TransD [14]	0.913	0.913
TransR [14]	0.902	0.904
TransH [14]	0.912	0.913
AttTucker	0.955	0.956

2.2. KG Negative Sampling

During the KG quality evaluation process, existing negative sampling techniques can generally be divided into two categories: techniques that disregard spurious-negative triples and those that consider spurious-negative triples [22–24]. Negative sampling techniques that disregard spurious-negative triples rely on the closed-world assumption, where all sampled triples are considered negative. Typical methods include Uniform Sampling [6], Self-Adversarial Sampling [25], RW-SANS [26], and NSCaching [27]. Among these, Uniform Sampling is the most basic negative sampling technique, where negative triples are sampled uniformly from the distribution. Self-Adversarial Sampling allocates varying degrees of importance to individual negative triples, thereby improving the model's learning performance. It introduces an adversarial mechanism when generating negative samples, modifying their sampling significance according to their scores, allowing the model to prioritize more challenging negative samples. RW-SANS samples negative triples from the graph structure through random walks, increasing the diversity and relevance of the sampled instances. To further improve sampling efficiency, NSCaching maintains a cache during the sampling process to store potential high-quality negative sample candidates, thereby reducing the computational cost of generating negative samples during each training iteration and enhancing both sampling efficiency and quality.

Conversely, techniques that consider spurious-negative triples aim to minimize the probability of incorrectly selecting spurious-negative samples during the sampling process. Typical techniques include CANS [28] and Bernoulli sampling [18]. CANS is a negative sampling method designed for KG embedding tasks, specifically addressing the issue of false-negative samples. By incorporating common-sense knowledge and the characteristics of complex relationships, CANS enhances the accuracy of negative sampling and reduces the probability of spurious-negative samples. Bernoulli sampling is a classic negative sampling method in KG embedding models, aimed at addressing the data distribution imbalance caused by complex relationships in KGs. By considering different relation patterns, Bernoulli sampling adjusts the sampling probability, effectively improving the quality of negative samples and enhancing the model's learning performance.

Compared to traditional negative sampling techniques, the negative sampling strategy introduced in this study offers significant advantages in regard to several aspects. Firstly, traditional negative sampling methods are often search-based and require screening through a large number of unlabeled negative samples to select high-quality ones,

thus increasing computational costs, and this process may not guarantee the diversity and difficulty of the negative samples. Secondly, methods like CANS rely on external common-sense information, which not only incurs high manual costs but also depends on the specific task, making it difficult to generalize. Although Bernoulli sampling avoids dependence on external information, it employs a fixed sampling scheme and lacks dynamic adjustment capability.

In contrast, the negative sampling strategy proposed in this paper does not merely avoid sampling false negatives. Instead, by introducing a spurious-negative sample optimization strategy, the sampled negative samples are optimized into high-quality training samples. This method dynamically filters and adjusts the mixing weights between negative samples during the sampling process, ensuring that the negative samples used in model training are not only representative but also provide effective supervisory signals. This strategy enhances the model's discriminative ability and robustness against noise, thereby improving performance in the quality evaluation task.

In summary, existing methods for knowledge graph quality evaluation and negative sampling techniques have made significant progress but still face certain challenges. This study aims to enhance the accuracy and robustness of knowledge graph quality evaluation by introducing an optimization strategy for spurious-negative triples that transforms sampled negative samples into high-quality training samples.

3. Methods

In this section, we propose a spurious-negative sample augmentation-based quality evaluation method for KGs. This method effectively identifies and handles spurious-negative samples through multi-scale spurious-negative triple detection and adaptive mixup based on attention mechanisms, thereby providing more precise supervisory signals during model training. Figure 1 illustrates the structure of the SNAQE model. We divided the SNAQE model into four main steps. Step 1 represents Negative Sampling, where negative triples are constructed by replacing either the head entity or the tail entity. Step 2 involves Spurious-Negative Triple Detection, in which negative triples are classified into spurious-negative triples and true-negative triples based on their multi-scale scores. Step 3 is the Adaptive Mixup Based on the Attention Mechanism, where suitable mixup partners are selected for each spurious-negative triple to achieve more precise sample generation. Step 4 is the Model Training phase, during which multiple scoring functions are utilized to train the model.

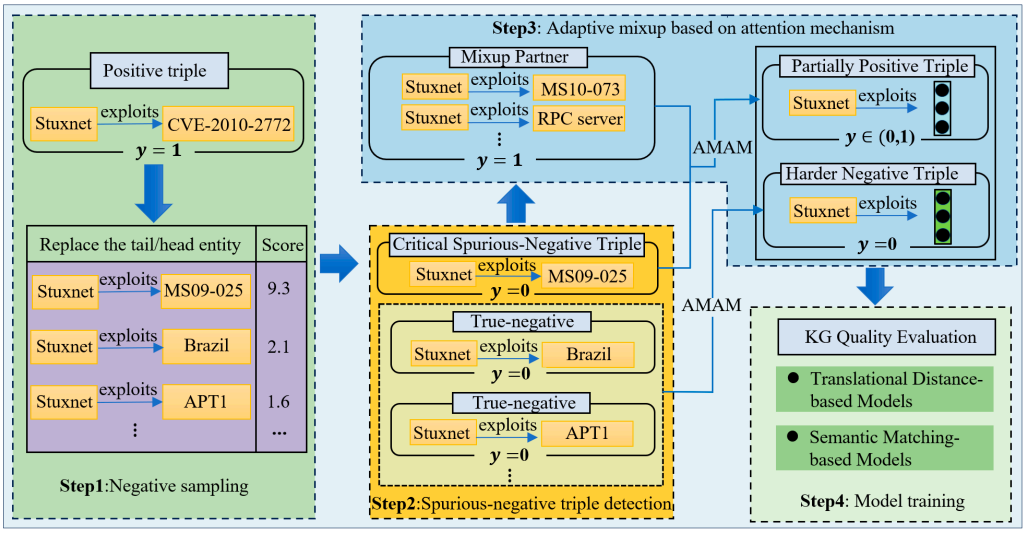


Figure 1. The structure of the SNAQE.

3.1. Multi-Scale Spurious-Negative Triple Detection

Formally, a knowledge graph can be represented as follows:

$$KG = (E, R, T), \tag{1}$$

where $E = \left\{e_1, e_2, \cdots, e_{|E|}\right\}$ is the set of entities, $R = \left\{r_1, r_2, \cdots, r_{|R|}\right\}$ is the set of relations, and $T \subseteq E \times R \times E$ is the set of triples. Each triple can be represented as (h, r, t), where h, r, and t represent the head entity, relation, and tail entity of the triple, respectively. For each triple (h, r, t), the corresponding k-dimensional embedding is represented as (v_h, v_r, v_t). The labeled dataset is denoted as $M = \left\{\left(h_j, r_j, t_j\right), y_i\right\}_{j=1}^{n_q+n_p}$, where n_q represents the number of positive samples labeled as $y = 1$, and n_p represents the number of negative samples labeled as $y = 0$. Typically, n_q equals the size of the factual set, i.e., $n_p = |T|$. As negative triples are absent from the KG, they are typically generated through a negative sampling approach.

During the process of negative sampling, we refer to the method in [18], where negative triples, T_N, are constructed by replacing h or t as follows:

$$T_N = \bigcup_{(h,r,t)\in T} \{(e, r, t) \notin T\} \cup \{(h, r, e) \notin T\}. \tag{2}$$

Based on the closed-world assumption, in KG-related tasks, most negative sampling techniques consider all triples that do not exist as negative instances. Therefore, the labeled triple dataset, M, can be represented as follows:

$$M = \{((h, r, t), y = 1) | (h, r, t) \in T\} \cup \left\{\left((h', r, t'), y = 0\right) \middle| \left(h', r, t'\right) \in T_N\right\}. \tag{3}$$

In this paper, we categorize the triples obtained through negative sampling into two groups based on the features of the triples: critical spurious-negative triples and true-negative triples. Critical spurious-negative triples are false-negative samples with scores that are nearly identical to those of positive samples. Considering that negative sampling in KGs involves replacing either the h or t of a triple, while the other components of the triple remain unchanged, we refer to the unchanged components as *primitive*. Therefore, we can define T_N as follows:

$$\begin{aligned} T_N = &\{(e, primitive) | e \in E, primitive = (r, t)\} \\ &\cup \{(primitive, e) | e \in E, primitive = (h, r)\}. \end{aligned} \tag{4}$$

The collection of positive samples where (h, r) and (r, t) serves as the primitive can be expressed as follows:

$$T^{hr}_{primitive} = \{(h, r, e_i) \in T | e_i \in E\}, \tag{5}$$

$$T^{rt}_{primitive} = \left\{\left(e_j, r, t\right) \in T \middle| e_j \in E\right\}. \tag{6}$$

The set of positive triples, T, is expressed as follows:

$$T = T^{hr}_{primitive} \cup T^{rt}_{primitive}. \tag{7}$$

To enhance the ability to identify critical spurious-negative samples, we introduce a local neighborhood score, which is measured based on the neighborhood structure of the entities in the KG. Specifically, for a given head entity, h', or tail entity, t', we define its local neighborhood, $N(h')$ or $N(t')$, as the collection of all entities and relations directly connected to it.

$$N\left(h'\right) = \left\{(r, t) \middle| \left(h', r, t\right) \in T\right\}, \tag{8}$$

$$N(t') = \{(h,r)|(h,r,t') \in T\}. \tag{9}$$

For a negative triple (h',r,t') to be detected, we first calculate the local neighborhood score of the head entity, h', and the tail entity, t'. The local neighborhood score of h' is defined as follows:

$$f_{loc}(h') = \frac{1}{|N(h')|} \sum_{(r,t_i)\in N(h')} \varphi(h',r,t_i). \tag{10}$$

Similarly, we can calculate the local neighborhood score of the tail entity, t', denoted as $f_{loc}(t')$. These local scores will be used to enhance the detection of spurious-negative samples, especially for relation types with complex local structures.

$$f_{loc}(t') = \frac{1}{|N(t')|} \sum_{(h_i,r)\in N(t')} \varphi(h_i,r,t'). \tag{11}$$

Based on this, we calculate the multi-scale spurious-negative sample score, $f_{MS\text{–}SNS}\left(h',r,t'\right)$, for spurious-negative samples:

$$f_{MS\text{–}SNS}\left(h',r,t'\right) = \alpha\varphi(h',r,t') + (1-\alpha)\left(\frac{f_{loc}(h') + f_{loc}(t')}{2}\right), \tag{12}$$

where $\varphi(h',r,t')$ is the global score, which is the score of the triple (h',r,t') given by the KG embedding model; $f_{loc}(h')$ and $f_{loc}(t')$ are the local neighborhood scores of the head entity and tail entity; and α is a hyperparameter controlling the weight between the global and local scores.

Next, we classify the negative triples with scores that closely resemble those of the positive samples under the same primitive as critical spurious-negative triples, denoted as T_{csn}. T_{csn} can be defined as follows:

$$T_{csn} = \left\{((h',r,t'),y=0)\middle|(h',r,t') \in T_N,\ \left[f^{T_{primitive}}\right]_{min} - \alpha_T \le f_{MS\text{–}SNS}\left(h',r,t'\right) \le \left[f^{T_{primitive}}\right]_{mean}\right\} \tag{13}$$

where $f^{T_{primitive}}$ denotes the set of scores for positive samples that have the identical primitive; $[Z]_{min}$ and $[Z]_{mean}$ represent the minimum and mean values of Z, respectively; and α_T is a hyperparameter that controls the estimation range during training.

Generally, when the number of elements in the positive triple set, $T_{primitive}$, with the same primitive is low, the probability of generating spurious-negative triples from this primitive is also low. Therefore, a threshold, τ, is defined to determine whether to derive T_{csn} from T_N:

$$T_N = \begin{cases} T_{\tilde{N}} & when\ \left|T_{primitive}\right| < \tau \\ T_{csn} \cup T_{\tilde{N}} & when\ \left|T_{primitive}\right| \ge \tau \end{cases}, \tag{14}$$

where $T_{\tilde{N}}$ is the set of true-negative samples.

The algorithmic process for multi-scale spurious-negative triple detection is detailed in Algorithm 1.

Algorithm 1. Multi-Scale Spurious-Negative Triple Detection

Input Knowledge graph, $KG = (E,R,T)$; embedding dimension, d; scoring function, f; threshold, τ.

Output: Spurious-negative triple set, T_{csn}, and true-negative triple set, $T_{\tilde{N}}$.

1: Initialize the embedding vectors of entity set, E, and relation set, R.

2: Generate the negative triple set, T_N:

$$T_N = \bigcup_{(h,r,t)\in T} \{(e,r,t) \notin T\} \cup \{(h,r,e) \notin T\}.$$

Algorithm 1. *Cont.*

3: Group the negative triples based on primitives and reorganize the negative triple set, T_N:

$$T_N = \{(e, primitive)|e \in E, primitive = (r,t)\}$$
$$\cup\{(primitive, e)|e \in E, primitive = (h,r)\}.$$

4: Compute the local neighborhood scores for the head entity, h', and tail entity, t':

$$f_{loc}(h') = \frac{1}{|N(h')|}\sum_{(r,t_i)\in N(h')}\varphi(h', r, t_i).$$

$$f_{loc}(t') = \frac{1}{|N(t')|}\sum_{(h_i,r)\in N(t')}\varphi(h_i, r, t').$$

5: Calculate the multi-scale spurious-negative triple score:

$$f_{MS\text{–}SNS}\left(h', r, t'\right) = \alpha\varphi(h', r, t') + (1-\alpha)\left(\frac{f_{loc}(h') + f_{loc}(t')}{2}\right).$$

6: Detect spurious-negative triples based on the scores in Step 5:

$$T_{csn} = \{\left((h', r, t'), y = 0\right)(h', r, t') \in T_N,$$
$$\left[f^{T_{primitive}}\right]_{min} - \alpha_T \leq f_{MS-SNS}\left(h', r, t'\right) \leq \left[f^{T_{primitive}}\right]_{mean}\}$$

7: Refine the final negative triple set based on the threshold, τ:

$$T_N = \begin{cases} T_{\tilde{N}} & when \ \ |T_{primitive}| < \tau \\ T_{cpn} \cup T_{\tilde{N}} & when \ \ |T_{primitive}| \geq \tau \end{cases}$$

8: Output: Spurious-negative triple set, T_{csn}, and true-negative triple set, $T_{\tilde{N}}$

3.2. Adaptive Mixup Based on Attention Mechanism (AMAM)

In the negative sampling process, directly labeling spurious-negative triples as positive ($y = 1$) would provide the model with incorrect supervision signals, which could negatively impact the model's training performance. To mitigate the interference caused by high-quality negative triples during model training, and building upon research in mixup techniques [29], we propose an attention mechanism-based adaptive mixup method. This method selects appropriate mixing partners for each negative sample triple to generate high-quality triples that encapsulate rich information.

First, we construct a potential positive set (T_{pps}) for each distinct primitive. During the model training process, T_{pps} is updated. The set T_{pps} includes the positive triples that are close to the model's current decision threshold.

$$\begin{aligned} T_{pps} = \{((h,r,t), y = 1)\big|(h,r,t) \in T_{primitive}, \\ f_{MS\text{–}SNS}(h,r,t) \leq \left[f^{T_{primitive}}\right]_{mean}\}. \end{aligned} \tag{15}$$

Essentially, critical spurious-negative triples, T_{csn}, have a high probability of being positive samples but are incorrectly labeled as negative samples. In this paper, we randomly select K mixing partners from T_{pps} for the critical spurious-negative triples and use an attention mechanism to construct partially positive triples. These partially positive triples

provide more accurate supervision signals to the model. Not only do they reduce the negative impact of spurious-negative triples on model training, but they also enhance the generalization ability of the model to some extent.

Moreover, for the true-negative triples, $T_{\sim N}$, we randomly select K real negative triples with the same primitive as mixing partners, thereby constructing higher-quality negative samples. It is worth noting that, as the construction of negative triples is performed under the same primitive, the true mixing occurs between the candidate entities and their corresponding counterparts. Let e_i and e_j represent the entities in the negative triples and their corresponding mixing partners. The range of mixing partners for negative triples can be represented as follows:

$$(e_i, y_j) \sim \begin{cases} Attention(T_{pps}) & if \quad (e_i, y_j) \in T_{csn} \\ Attention\left(T_{\sim N}\right) & if \quad (e_i, y_j) \in T_{\sim N} \end{cases}. \tag{16}$$

In the mixing operation, to enhance the rationality and flexibility of the mixing process, we combine the attention mechanism to dynamically adjust the mixing weights. First, for each negative sampled triple and the candidate positive or negative samples, we construct the Q, K, and V matrices:

$$\begin{aligned} Q_i &= W_Q e_i, \\ K_j &= W_K e_j, \\ V_j &= W_V e_j. \end{aligned} \tag{17}$$

For a spurious-negative sample, e_i, and each candidate sample, e_j (where $j = 1, 2, \ldots, K$), we compute the similarity, s_{ij}, using the dot product of Q and K:

$$s_{ij} = \frac{Q_i\left(K_j\right)^T}{\sqrt{d_k}}, \tag{18}$$

where $\frac{1}{\sqrt{d_k}}$ is the scaling factor used to prevent the dot product from becoming too large and affecting gradient propagation.

After calculating the similarity, s_{ij}, we normalize the similarity using the softmax function to obtain the attention weights, α_{ij}:

$$\alpha_{ij} = \frac{\exp(s_{ij})}{\sum_{j=1}^{K} \exp(s_{ij})}. \tag{19}$$

Using the calculated attention weights, α_{ij}, we combine the multiple candidate samples to generate the new sample embedding, $\hat{e}_i$, and label, $\hat{y}_i$:

$$\hat{e}_i = \sum_{j=1}^{K} \alpha_{ij} e_j + \left(1 - \sum_{j=1}^{K} \alpha_{ij}\right) e_i, \tag{20}$$

$$\hat{y}_i = \sum_{j=1}^{K} \alpha_{ij} y_j. \tag{21}$$

This adaptive mixing strategy reasonably allocates the weights between samples through the attention mechanism, making the mixing process more flexible. Through adaptive mixing, the model can generate partially positive samples and higher-quality negative samples, thereby providing more precise supervision signals for the training process. This significantly enhances the performance of the model in the quality evaluation task.

3.3. Model Training

Throughout the training process, to ensure that our model can comprehensively and effectively capture the rich information and complex relationships embedded in the KG,

we utilize various scoring functions from knowledge representation learning to train the model, including ComplEx, DistMult, TransE, and RotatE, as shown in Table 2.

Table 2. Scoring functions, parameters, and time complexity for different models.

Model	Scoring Function	Parameters	Time Complexity
ComplEx	$Re(\langle \boldsymbol{v_r}, \boldsymbol{v_h}, \overline{\boldsymbol{v_t}} \rangle)$	$\boldsymbol{v_h}, \boldsymbol{v_r}, \boldsymbol{v_t} \in \mathbb{C}^k$	$\mathcal{O}(k)$
DistMult	$\langle \boldsymbol{v_r}, \boldsymbol{v_h}, \boldsymbol{v_t} \rangle$	$\boldsymbol{v_h}, \boldsymbol{v_r}, \boldsymbol{v_t} \in \mathbb{R}^k$	$\mathcal{O}(k)$
TransE	$-\|(\boldsymbol{v_h} + \boldsymbol{v_r}) - \boldsymbol{v_t}\|_p$	$\boldsymbol{v_h}, \boldsymbol{v_r}, \boldsymbol{v_t} \in \mathbb{R}^k$	$\mathcal{O}(k)$
RotatE	$-\|\boldsymbol{v_h} \circ \boldsymbol{v_r} - \boldsymbol{v_t}\|^2$	$\boldsymbol{v_h}, \boldsymbol{v_r}, \boldsymbol{v_t} \in \mathbb{C}^k, \|\boldsymbol{v_{ri}}\| = 1$	$\mathcal{O}(k)$

The SNAQE model, employing various scoring functions, is referred to as SNAQE-ComplEx, SNAQE-DistMult, SNAQE-TransE, and SNAQE-RotatE. It also incorporates a self-adversarial negative sampling strategy for loss calculation and model optimization [25]:

$$L = -\log\sigma(\omega + \varphi(h, r, t)) - \sum_{i=1}^{n} p\left(h_i', r, t_i'\right) \log\sigma\left(-\omega - \varphi\left(h_i', r, t_i'\right)\right), \tag{22}$$

where n represents the overall count of negative samples, σ refers to the sigmoid function, and ω denotes the fixed margin. The self-adversarial weight, $p\left(h_i', r, t_i'\right)$, for the negative triple is computed as follows:

$$p\left(h_j', r, t_j' \middle| \{(h_i, r_i, t_i)\}\right) = \frac{\exp \vartheta \varphi\left(h_j', r, t_j'\right)}{\sum_i \exp \vartheta \varphi\left(h_i', r, t_i'\right)}, \tag{23}$$

where ϑ denotes the sampling temperature.

4. Results and Discussion

This section evaluates the effectiveness of the SNAQE model on general-domain datasets through detailed comparative experiments, while also examining its quality assessment performance on cybersecurity-domain datasets.

4.1. Dataset Description

We first evaluated the performance of our model on the widely used general-domain datasets, FB15K and WN18, to assess its robustness and scalability across different datasets. Subsequently, the performance of the model was tested on the cybersecurity dataset CS13K. Table 3 summarizes the dataset statistics.

Table 3. Statistical characteristics of the datasets.

Dataset	Entities	Relations	Training Set	Validation Set	Test Set
FB15K	14,951	1345	483,142	50,000	59,071
WN18	40,943	18	141,442	5000	5000
CS13K	4494	12	10,000	/	3027

(1) FB15K [30]: This is a general-domain KG dataset that includes 14,951 entities and 1345 types of relations. The dataset's characteristics include its large scale and rich set of knowledge facts.

(2) WN18 [7]: This dataset contains 18 relation types and 40,943 entities. It is a subset extracted from the WordNet dataset and is widely used in the field of vocabulary relation inference in KGs.

(3) CS13K [31]: This is a cybersecurity KG dataset that includes 12 relation types, 4494 entities, and 13,027 triples.

4.2. Experimental Settings and Evaluating Metrics

During the KG quality evaluation, triples are categorized according to their final predicted probability, p. When $p \geq 0.5$, the model classifies the triple as a positive sample; when $p < 0.5$, it classifies the triple as a negative sample. The evaluation metrics used are F1 score and accuracy. Table 4 presents the main parameter settings of the SNAQE model.

$$Precision = \frac{TP}{TP + FP}, \tag{24}$$

$$Recall = \frac{TP}{TP + FN}, \tag{25}$$

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}, \tag{26}$$

$$F1 = \frac{2Precision * Recall}{Precision + Recall}, \tag{27}$$

where TP (true positive) refers to the positive examples correctly classified as positive, FN (false negative) refers to the positive examples incorrectly classified as negative, FP (false positive) refers to the negative examples incorrectly classified as positive, and TN (true negative) refers to the negative examples correctly classified as negative.

Table 4. The key parameter configurations of the SNAQE model.

Dimension embeddings	100
Batch size	512
Epochs	2000
Threshold value, τ	3
Learning rate	0.001
The number of K	5
Hyperparameter α_T	3

4.3. Experimental Results and Analysis

4.3.1. Performance of the Model on General-Domain KG Datasets

Within this part of the study, we first evaluated the performance of the SNAQE model on the FB15K and WN18 datasets and compared it with existing negative sampling methods, including Uniform Sampling, Self-Adversarial Sampling, RW-SANS, NSCaching, CANS, and Bernoulli Sampling. The results of the experiments are presented in Tables 5–8.

On the FB15K dataset, the SNAQE model demonstrates significant performance improvements across various Translational Distance-Based Models and Semantic Matching-Based Models. Specifically, SNAQE-TransE achieves accuracy and F1-score values of 0.916 and 0.919, respectively, showing an approximately 1% gain compared to the second-best method, CANS-TransE. This indicates that the SNAQE method enhances the model's ability to represent triples by effectively handling spurious-negative samples. The SNAQE-RotatE model demonstrates exceptional performance, attaining an F1 score of 0.969 on the FB15K dataset, significantly surpassing that of other methods. This demonstrates that, in Translational Distance-Based Models, the SNAQE-based model can further improve the KG evaluation performance by effectively identifying spurious-negative samples and constructing high-quality training samples. In Semantic Matching-Based Models, SNAQE-DistMult and SNAQE-ComplEx achieve F1 scores of 0.858 and 0.917, respectively, outperforming other negative sampling methods using the same scoring functions. For instance,

SNAQE-ComplEx improves accuracy by about 3.5% compared to the second-best method, NSCaching-ComplEx. The SNAQE model also demonstrates excellent performance when applied to the WN18 dataset. SNAQE-RotatE achieves an F1 score of 0.948, exceeding CANS-RotatE's 0.932. Compared to other negative sampling methods, SNAQE-DistMult and SNAQE-ComplEx also show remarkable enhancements in both accuracy and F1 score, highlighting the superiority of the SNAQE method.

Table 5. Quality evaluation performance of different negative sampling strategies on TransE.

Translational Distance-Based Models	FB15K		WN18	
	Accuracy	F1 Score	Accuracy	F1 Score
Disregard the influence of false-negative samples				
Uniform-TransE	0.868	0.876	0.852	0.853
Self-Adv-TransE	0.895	0.897	0.872	0.875
RW-SANS-TransE	0.902	0.905	0.878	0.881
NSCaching-TransE	0.881	0.884	0.864	0.866
Consider the influence of false-negative samples				
CANS-TransE	0.907	0.910	0.888	0.889
Bernoulli-TransE	0.901	0.902	0.879	0.882
Ours				
SNAQE-TransE	0.916	0.919	0.893	0.896

Table 6. Quality evaluation performance of different negative sampling strategies on RotatE.

Translational Distance-Based Models	FB15K		WN18	
	Accuracy	F1 Score	Accuracy	F1 Score
Disregard the influence of false-negative samples				
Uniform-RotatE	0.932	0.934	0.922	0.923
Self-Adv-RotatE	0.929	0.930	0.916	0.918
RW-SANS-RotatE	0.943	0.945	0.925	0.928
NSCaching-RotatE	0.944	0.946	0.929	0.931
Consider the influence of false-negative samples				
CANS-RotatE	0.949	0.951	0.931	0.932
Bernoulli-RotatE	0.935	0.936	0.923	0.925
Ours				
SNAQE-RotatE	0.967	0.969	0.945	0.948

Additionally, we examined how the proportion of noise within triples influences the effectiveness of the SNAQE model. We evaluated the F1 scores on the FB15K and WN18 datasets, with the results shown in Figure 2. From the experimental data, it can be observed that on the FB15K dataset, as the noise triple ratio gradually decreases from 50% (N50) to 10% (N10), the F1 score of the SNAQE-based models remains relatively stable. This indicates that the model demonstrates strong robustness when handling varying noise ratios, confirming the advantage of the adaptive mixup based on attention mechanism in reducing the interference from noise triples, alleviating the negative impact of noise disturbances, and thus improving the overall model performance. For the WN18 dataset, the SNAQE model also performs exceptionally well across different noise ratios, further validating that the model can maintain stable quality evaluation performance even under varying noise conditions.

Table 7. Quality evaluation performance of different negative sampling strategies on DistMult.

Semantic Matching-Based Models	FB15K		WN18	
	Accuracy	F1 Score	Accuracy	F1 Score
Disregard the influence of false-negative samples				
Uniform-DistMult	0.808	0.810	0.803	0.806
Self-Adv-DistMult	0.817	0.818	0.812	0.815
RW-SANS-DistMult	0.836	0.838	0.825	0.828
NSCaching-DistMult	0.848	0.851	0.838	0.839
Consider the influence of false-negative samples				
CANS-DistMult	0.843	0.845	0.831	0.834
Bernoulli-DistMult	0.841	0.842	0.828	0.831
Ours				
SNAQE-DistMult	0.857	0.858	0.843	0.846

Table 8. Quality evaluation performance of different negative sampling strategies on ComplEx.

Semantic Matching-Based Models	FB15K		WN18	
	Accuracy	F1 Score	Accuracy	F1 Score
Disregard the influence of false-negative samples				
Uniform-ComplEx	0.818	0.820	0.813	0.816
Self-Adv-ComplEx	0.830	0.831	0.825	0.828
RW-SANS-ComplEx	0.861	0.863	0.851	0.853
NSCaching-ComplEx	0.883	0.886	0.877	0.878
Consider the influence of false-negative samples				
CANS-ComplEx	0.875	0.878	0.869	0.871
Bernoulli-ComplEx	0.869	0.872	0.865	0.868
Ours				
SNAQE-ComplEx	0.915	0.917	0.905	0.908

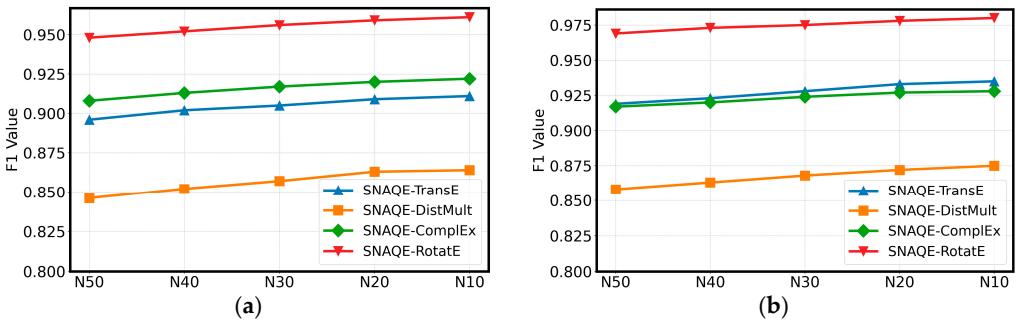


Figure 2. Influence of noise triple ratio on model performance: (**a**) FB15K and (**b**) WN18.

Furthermore, a visualization of the quality evaluation outcomes of the SNAQE-RotatE model using the FB15K dataset was conducted to further confirm its effectiveness in differentiating positive and negative triples. Figure 3 shows the confidence scores for positive and negative triples evaluated by the SNAQE-RotatE model with the FB15K dataset. The visualization reveals that the confidence scores for positive triples predominantly fall within the range from 0.85 to 1.0, whereas those for negative triples are primarily distributed between 0 and 0.1. This distribution substantiates the capability of SNAQE-based models in regard to the quality assessment of KGs in general domains.

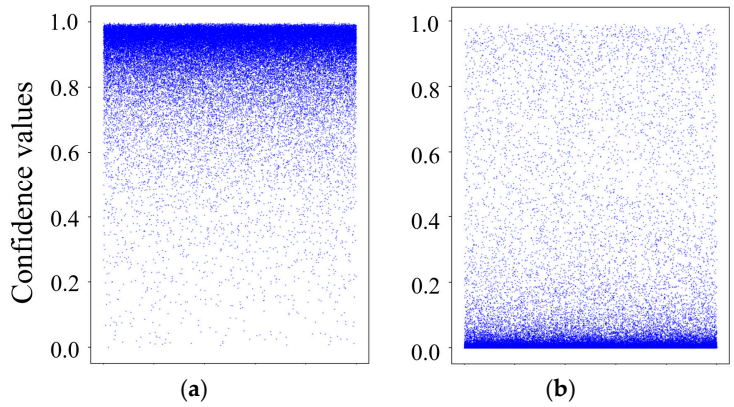


Figure 3. Display of quality evaluation outcomes: (**a**) positive triples and (**b**) negative triples.

To validate the model's ability to balance the identification and classification of positive and negative triples in the quality evaluation task, we also analyzed the recall and precision of the SNAQE model during quality evaluation. Figure 4 presents the recall–precision curve of the SNAQE-RotatE model on the FB15K dataset.

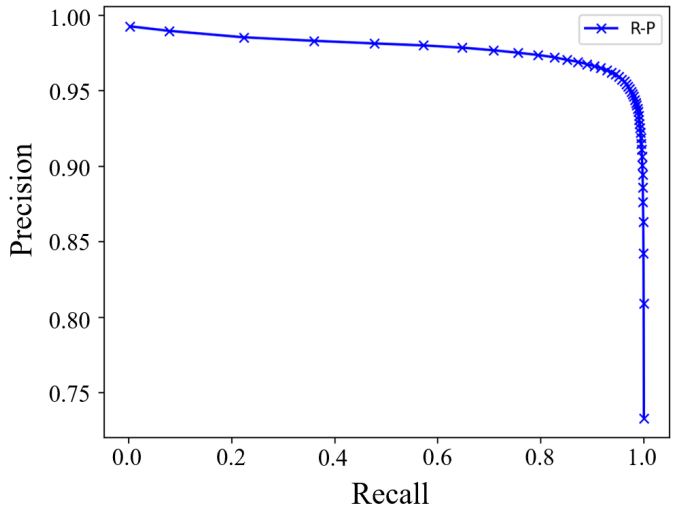


Figure 4. Recall–precision curve.

As depicted in Figure 4, for recall values under 0.85, the model maintains a high and steady level of precision. However, once the recall exceeds 0.85 and continues to rise, the precision starts to drop rapidly. This phenomenon may arise from the model increasingly misclassifying negative triples as positive ones at higher recall levels, increasing the error rate and thus lowering the overall precision. It is noteworthy that when the recall reaches approximately 0.96, the model achieves an optimal balance, maintaining a high precision while ensuring a high recall. This observation indicates that the SNAQE-RotatE model can effectively control noise interference on the FB15K dataset and achieve a good balance between recall and precision, further confirming the effectiveness and reliability of the method in quality evaluation tasks for general-domain KGs.

4.3.2. Performance of the Model on Cybersecurity Dataset

We also evaluated the performance of the PNEQA-based model on the cybersecurity KG dataset to assess its effectiveness in a specific domain. The outcomes, shown in Tables 9 and 10, demonstrate that the SNAQE model outperforms other negative sampling methods in regard to both accuracy and F1 score.

Table 9. Evaluation results of different negative sampling methods on Translational Distance-Based Models.

Translational Distance-Based Models	CS13K	
	Accuracy	F1 Score
Disregard the influence of false-negative samples		
Uniform-RotatE	0.901	0.904
Self-Adv-RotatE	0.899	0.901
RW-SANS-RotatE	0.904	0.907
NSCaching-RotatE	0.907	0.911
Consider the influence of false-negative samples		
CANS-RotatE	0.914	0.915
Bernoulli-RotatE	0.911	0.913
Ours		
SNAQE-RotatE	0.926	0.928

Table 10. Evaluation results of different negative sampling methods on Semantic Matching-Based Models.

Semantic Matching-Based Models	CS13K	
	Accuracy	F1 Score
Disregard the influence of false-negative samples		
Uniform-ComplEx	0.851	0.853
Self-Adv-ComplEx	0.861	0.862
RW-SANS-ComplEx	0.880	0.884
NSCaching-ComplEx	0.889	0.892
Consider the influence of false-negative samples		
CANS-ComplEx	0.878	0.881
Bernoulli-ComplEx	0.881	0.883
Ours		
SNAQE-ComplEx	0.901	0.903

In Translational Distance-Based Models, when the influence of spurious-negative triples was ignored, the best F1 score of 0.911 was achieved by the NSCaching-RotatE model. However, when spurious-negative triples are considered, the model's quality assessment performance improves noticeably. This demonstrates that spurious-negative triples play a significant role in the quality assessment process. Notably, our proposed SNAQE-RotatE achieved an F1 score of 0.928, which is significantly higher than other models.

In Semantic Matching-Based Models, similar results were observed. Compared to other models, the SNAQE-ComplEx model achieved the best evaluation results, validating the effectiveness and superiority of SNAQE-based models in assessing the quality of cybersecurity KGs.

To further analyze the performance of the SNAQE-based model in quality evaluations across different relation types, we evaluated individual relation categories within the CS13K dataset and reported the model's precision, recall, and F1 score. Table 11 presents the quality evaluation results of the SNAQE-RotatE model for various relation types.

As shown in Table 11, the SNAQE-RotatE model demonstrates varying levels of performance across distinct categories of relations. For instance, the model achieves higher recall and precision for relations like "hasAttackLocation" and "hasVulnerability", indicating good identification capability for these relations. In contrast, relations like "indicates"

and "belongTo" exhibit relatively lower precision and recall values. This variation can be influenced by several factors. First, there is an imbalance in the number of data samples for different relation types. More common relations, which occupy a larger proportion of the training data, allow the model to learn their features more thoroughly, resulting in higher precision and recall. Conversely, for relation types with fewer samples, the model's recognition ability is limited, leading to lower evaluation metrics. Second, the semantic complexity and feature differences between relation types also affect the performance of the model. Some relations may have clear distinguishing features, enabling the model to accurately capture and differentiate their semantics, while others may be semantically complex or share a higher similarity, increasing the difficulty of recognition.

Table 11. Assessment outcomes for different relation categories on CS13K dataset.

Relations	Precision	Recall	F1 Score
Exploits	0.916	0.899	0.907
Cause	0.926	0.929	0.927
BelongTo	0.873	0.878	0.875
HasCharacteristics	0.923	0.926	0.924
HasAttackLocation	0.952	0.958	0.955
HasVulnerability	0.953	0.958	0.955
HasAttackTime	0.953	0.961	0.957
Targets	0.968	0.963	0.965
Associate	0.920	0.926	0.923
Mitigates	0.922	0.927	0.924
Indicates	0.903	0.906	0.904
Use	0.913	0.921	0.917
Average	0.927	0.929	0.928

The quality evaluation for specific relation types further reveals the applicability of the SNAQE-based models in cybersecurity KGs. This analysis not only helps us understand the model's limitations across various types of relations but also provides guidance for optimizing cybersecurity KGs.

4.4. Case Study

To validate the performance of the SNAQE model in quality assessment on the cybersecurity knowledge graph dataset CS13K, we conducted a detailed case study. In the experiment, we selected a primitive $(h, r) = (\text{Stuxnet}, \text{exploits})$ and analyzed how the model optimizes spurious-negative samples and enhances evaluation performance through comprehensive experiments and visualizations.

In the experiment, we randomly selected 20 positive triples from the training set, 10 spurious-negative triples, and 10 true-negative triples from the validation and test sets. The selected primitive $(h, r) = (\text{Stuxnet}, \text{exploits})$, where "Stuxnet" is the head entity, and "exploits" is the relation. "Stuxnet" represents a typical network worm widely regarded as an advanced cyberattack tool targeting industrial control systems, while "Exploits" represent the relations between Stuxnet and the vulnerabilities or attack methods it utilizes during its propagation and attack process. We employed 2D t-SNE dimensionality reduction to map the high-dimensional tail entity embeddings into a two-dimensional space, aiming to verify the capability of the SNAQE model in identifying and optimizing spurious-negative samples. The results are shown in Figure 5, where Figure 5a represents the experimental results without spurious-negative triple detection, and Figure 5b represents the results with spurious-negative triple detection. A comparison of Figure 5a,b demonstrates that the SNAQE model effectively moves the embeddings of spurious-negative triples closer to the distribution region of positive triples while pushing true-negative triples farther

from the positive distribution region. The analysis of tail entity embeddings under the specific primitive (Stuxnet, exploits) reveals that without spurious-negative triple detection, the model erroneously treats spurious-negative triples as true-negative triples, thereby affecting the training effectiveness to some extent. In contrast, with spurious-negative triple detection, the model shifts the embeddings of "false negatives" closer to the positive triple distribution region, thereby enhancing evaluation performance. The significant differences in embedding distributions indicate that the proposed SNAQE model effectively mitigates the adverse effects of spurious-negative samples on model training and improves the reliability and accuracy of knowledge graph quality evaluation.

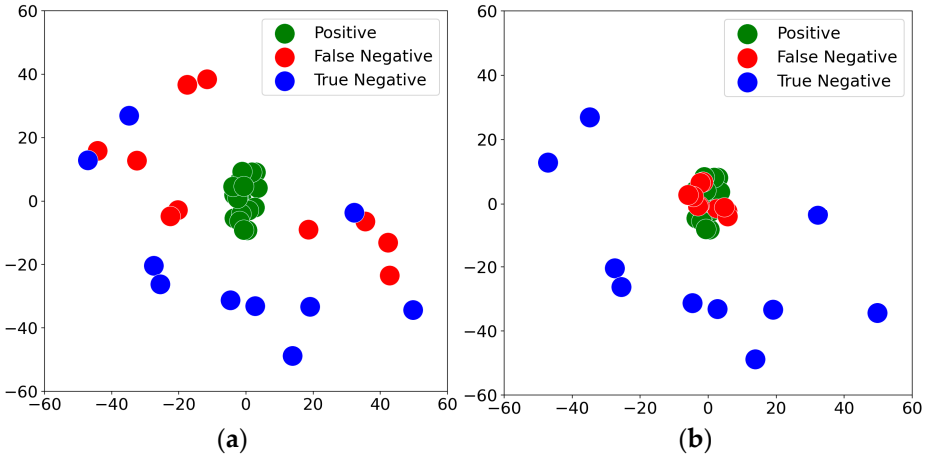


Figure 5. Two-dimensional t-SNE visualization of tail entity embeddings for the same primitive: (**a**) w/o SNAQE and (**b**) SNAQE.

4.5. Ablation Study

To validate the contribution of each component in the SNAQE model, we conducted ablation studies to assess the impact of key components. Table 12 presents the experimental outcomes for the SNAQE-RotatE model.

Table 12. Ablation study on the SNAQE model.

Dataset	Models	Accuracy	F1 Score
FB15K	SNAQE-RotatE	0.967	0.969
	w/o SNT	0.932	0.934
	w/o LocalN	0.953	0.955
	RandomM	0.949	0.950
CS13K	SNAQE-RotatE	0.926	0.928
	w/o SNT	0.895	0.896
	w/o LocalN	0.913	0.915
	RandomM	0.910	0.911

First, we evaluated the impact of ignoring spurious-negative triples (w/o SNT). When the labels of spurious-negative triples were directly assigned a value of 0, the performance of the model significantly decreased. For the FB15K dataset, omitting the SNT module led to a 3.6% reduction in the F1 score compared to the SNAQE-RotatE model; for the CS13K dataset, the F1 score dropped from 0.928 to 0.896. These results highlight the importance of identifying and optimizing spurious-negative triples. By specifically handling spurious-negative triples, the model can reduce the impact of noisy data and significantly improve F1 score and accuracy in quality evaluation tasks.

Next, we evaluated the contribution of local neighborhood information to the quality evaluation results. After removing local neighborhood information (w/o LocalN), the performance of the model declined. For the FB15K dataset, the accuracy dropped from 0.967 to 0.953, and the F1 score dropped from 0.969 to 0.955. On the CS13K dataset, the F1 score dropped from 0.928 to 0.915. These results demonstrate that the inclusion of local neighborhood information effectively enhances the performance of the model.

Additionally, we validated the impact of different mixing strategies in the adaptive mixup module based on the attention mechanism. We replaced the adaptive mixup strategy with a random mixing strategy (RandomM). The results showed that removing the attention-based adaptive mixup module led to a decrease in model performance. For the FB15K dataset, the F1 score dropped by 3.5%, and for the CS13K dataset, the F1 score dropped from 0.928 to 0.911. These results confirm that the attention-based adaptive mixup strategy can dynamically adjust the mixing weights according to the features of the samples, thereby improving the performance of the model in quality evaluation tasks.

5. Conclusions and Future Works

This paper proposes a spurious-negative sample augmentation-based method for quality evaluation of cybersecurity KGs. The method comprises two key modules: multi-scale spurious-negative triple detection and attention mechanism-based adaptive mixing. The multi-scale spurious-negative triple detection module effectively distinguishes spurious-negative triples from true-negative samples during the negative sampling process, thereby alleviating the noise issues caused by traditional negative sampling methods. The adaptive mixup module utilizes an attention mechanism to select appropriate mixup targets for each spurious-negative sample, generating more precise partial positive samples in the embedding space, thus further enhancing the model's training effectiveness. Extensive experimental results demonstrate that the SNAQE model demonstrates excellent performance in both general and cybersecurity domain KG quality evaluations, significantly improving the accuracy and F1 score of the model. Notably, it achieves State-of-the-Art results on the FB15K dataset. Despite the notable progress achieved by SNAQE in KG quality evaluation, there is still potential for further enhancement. Future research could explore more diverse negative sampling strategies to improve the model's ability to distinguish a wider range of spurious-negative samples. Additionally, incorporating more advanced knowledge representation learning models could enable the capture of more complex structural and semantic information in KGs, further enhancing the accuracy of quality evaluation and the applicability of the model.

Author Contributions: Conceptualization, B.C. and Z.S.; methodology, B.C., Z.S., and H.L.; investigation, B.C.; writing—original draft preparation, B.C.; writing—review and editing, Z.S.; visualization, B.C.; supervision, Z.S. and H.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China (Grant no. 61771001).

Data Availability Statement: Publicly available datasets were analyzed in this research. The FB15K dataset can be found here (https://huggingface.co/datasets/VLyb/FB15k; accessed on 1 January 2023). The WN18 dataset can be found here (https://huggingface.co/datasets/VLyb/WN18; accessed on 1 January 2023). The CS13K dataset that supports the findings of this study is available from the corresponding author upon reasonable request.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Jia, Y.; Qi, Y.; Shang, H.; Jiang, R. A practical approach to constructing a knowledge graph for cybersecurity. *Engineering* **2018**, *4*, 53–60. [CrossRef]
2. Zhao, X.; Jiang, R.; Han, Y.; Li, A.; Peng, Z. A survey on cybersecurity knowledge graph construction. *Comput. Secur.* **2024**, *136*, 103524. [CrossRef]
3. Li, H.; Shi, Z.; Pan, C.; Zhao, D.; Sun, N. Cybersecurity knowledge graphs construction and quality assessment. *Complex Intell. Syst.* **2024**, *10*, 1201–1217. [CrossRef]
4. Mendes, P.; Mühleisen, H.; Bizer, C. Sieve: Linked data quality assessment and fusion. In Proceedings of the 2012 Joint EDBT/ICDT Workshops, Berlin, Germany, 30 March 2012; pp. 116–123.
5. Shi, Z.; Li, H.; Zhao, D.; Pan, C. Research on quality assessment methods for cybersecurity knowledge graphs. *Comput. Secur.* **2024**, *142*, 103848. [CrossRef]
6. Bordes, A.; Usunier, N.; Garcia, D. Translating embeddings for modeling multi-relational data. In *Advances in Neural Information Processing Systems, Proceedings of the 27th Annual Conference on Neural Information Processing Systems 2013, Lake Tahoe, NV, USA, 5–8 December 2013*; Neural Information Processing Systems Foundation, Inc.: La Jolla, CA, USA, 2013; Volume 26.
7. Dettmers, T.; Minervini, P.; Stenetorp, P. Convolutional 2D knowledge graph embeddings. In Proceedings of the AAAI Conference on Artificial Intelligence 2018, New Orleans, LA, USA, 2–7 February 2018; Volume 32.
8. Kadlec, R.; Bajgar, O.; Kleindienst, J. Knowledge base completion: Baselines strike back. *arXiv* **2017**, arXiv:1705.10744.
9. Sha, Y.; Feng, Y.; He, M.; Liu, S.; Ji, Y. Retrieval-Augmented Knowledge Graph Reasoning for Commonsense Question Answering. *Mathematics* **2023**, *11*, 3269. [CrossRef]
10. Zeng, Z.; Cheng, Q.; Si, Y. Logical Rule-Based Knowledge Graph Reasoning: A Comprehensive Survey. *Mathematics* **2023**, *11*, 4486. [CrossRef]
11. Cao, J.; Fang, J.; Meng, Z.; Liang, S. Knowledge graph embedding: A survey from the perspective of representation spaces. *ACM Comput. Surv.* **2024**, *56*, 1–42. [CrossRef]
12. Xie, R.; Liu, Z.; Lin, F. Does william shakespeare really write hamlet? knowledge representation learning with confidence. In Proceedings of the AAAI Conference on Artificial Intelligence 2018, New Orleans, LA, USA, 2–7 February 2018; Volume 32.
13. Zhang, Y.; Yao, Q.; Dai, W. AutoSF: Searching scoring functions for knowledge graph embedding. In Proceedings of the 2020 IEEE 36th International Conference on Data Engineering (ICDE), Dallas, TX, USA, 20–24 April 2020; IEEE: New York, NY, USA, 2020; pp. 433–444.
14. Jia, S.; Xiang, Y.; Chen, X. Triple trustworthiness measurement for knowledge graph. In Proceedings of the World Wide Web Conference, San Francisco, CA, USA, 13–17 May 2019; pp. 2865–2871.
15. Bougiatiotis, K.; Fasoulis, R.; Aisopos, F. Guiding graph embeddings using path-ranking methods for error detection innoisy knowledge graphs. *arXiv* **2020**, arXiv:2002.08762.
16. Yang, B.; Yih, W.; He, X. Embedding entities and relations for learning and inference in knowledge bases. *arXiv* **2014**, arXiv:1412.6575.
17. Trouillon, T.; Dance, C.; Gaussier, É. Knowledge graph completion via complex tensor factorization. *J. Mach. Learn. Res.* **2017**, *18*, 1–38.
18. Wang, Z.; Zhang, J.; Feng, J. Knowledge graph embedding by translating on hyperplanes. In Proceedings of the AAAI Conference on Artificial Intelligence, Québec City, QC, Canada, 27–31 July 2014; Volume 28.
19. Lin, Y.; Liu, Z.; Sun, M. Learning entity and relation embeddings for knowledge graph completion. In Proceedings of the AAAI Conference on Artificial Intelligence, Austin, TX, USA, 25–30 January 2015; Volume 29.
20. Balažević, I.; Allen, C.; Hospedales, T. Tucker: Tensor factorization for knowledge graph completion. *arXiv* **2019**, arXiv:1901.09590.
21. Zhao, Y.; Feng, H.; Gallinari, P. Embedding learning with triple trustiness on noisy knowledge graph. *Entropy* **2019**, *21*, 1083. [CrossRef]
22. Che, F.; Tao, J. M2ixKG: Mixing for harder negative samples in knowledge graph. *Neural Netw.* **2024**, *177*, 106358. [CrossRef] [PubMed]
23. Zhang, J.; Li, Y.; Zou, R.; Zhang, J.; Jiang, R.; Fan, Z.; Song, X. Hyper-relational knowledge graph neural network for next POI recommendation. *World Wide Web* **2024**, *27*, 46. [CrossRef]
24. Chen, X.; Zhang, W.; Yao, Z.; Chen, M.; Tang, S. Negative sampling with adaptive denoising mixup for knowledge graph embedding. In Proceedings of the International Semantic Web Conference, Athens, Greece, 6–10 November 2023; pp. 253–270.
25. Sun, Z.; Deng, Z.; Nie, J. Rotate: Knowledge graph embedding by relational rotation in complex space. *arXiv* **2019**, arXiv:1902.10197.
26. Ahrabian, K.; Feizi, A.; Salehi, Y. Structure aware negative sampling in knowledge graphs. *arXiv* **2020**, arXiv:2009.11355.
27. Zhang, Y.; Yao, Q.; Shao, Y. NSCaching: Simple and efficient negative sampling for knowledge graph embedding. In Proceedings of the 2019 IEEE 35th International Conference on Data Engineering (ICDE), Macau, China, 8–11 April 2019; IEEE: New York, NY, USA, 2019; pp. 614–625.

28. Niu, G.; Li, B.; Zhang, Y. CAKE: A scalable commonsense-aware framework for multi-view knowledge graph completion. *arXiv* **2022**, arXiv:2202.13785.
29. Huang, T.; Dong, Y.; Ding, M. Mixgcf: An improved training method for graph neural network-based recommender systems. In Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining, Virtual, 14–18 August 2021; pp. 665–674.
30. Balažević, I.; Allen, C.; Hospedales, T. Hypernetwork knowledge graph embeddings. In Proceedings of the Artificial Neural Networks and Machine Learning–ICANN 2019: Workshop and Special Sessions, Munich, Germany, 17–19 September 2019; Springer International Publishing: Cham, Switzerland, 2019; pp. 553–565.
31. Shi, Z.; Li, H.; Zhao, D. Research on Relation Classification Tasks Based on Cybersecurity Text. *Mathematics* **2023**, *11*, 2598. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.