Mitigating Membership Inference Attacks on Knowledge Graphs

Jnaini Abdellah(✉)

Department of Computer Science, Mohamed First University, Oujda 60000, Morocco
abdellah.jnaini@ump.ac.ma

Abstract. Knowledge Graphs (KGs) are structured representations that model entities and their relationships in a connected form. By leveraging their capabilities in the link prediction task and node classification on graphs, They have been successfully used in a variety of fields, including drug discovery, community detection, and location sharing applications. The usage of KGs in privacy-sensitive contexts has been made easier by these potent uses and the wide availability of graphs in various domains. Unfortunately, KGs are susceptible to well-defined attacks that could reveal private information. Membership inference attacks, which enable the attacker to determine if a certain sample was included in the training dataset, are our primary area of interest. In order to avoid this, we present three LLM-guided defense techniques that are implemented at the posterior level: knowledge distillation, secure aggregation, and posterior encoding with noise. Our suggested methods preserve the model's performance on the link prediction task while successfully reducing MIA accuracy. Our results provide insights into striking a balance between privacy preservation and predictive performance, and they have been confirmed through extensive tests on popular KG designs.

1 Introduction

The recent accomplishments of Deep Neural Networks (DNNs) have revolutionized the field of Machine Learning (ML), especially in machine translation [1], speech recognition [2], and object detection [3] tasks. This revolution is ascribed to the progress of Deep Learning (DL) architectures, such as Convolutional Neural Networks (CNNs) [4], Recurrent Neural Networks (RNNs) [5], and Autoencoders [6]. The input representation is obtained concurrently with the downstream task within the deep learning architecture. The process of collaborative learning requires no manual feature development because it operates differently from traditional machine learning methods. The processing power improvements have enabled better performance of neural architectures through the development of advanced and more intricate systems. The extensive training datasets which researchers now have access to, enable better learning and generalization capabilities, which lead to improved performance across different tasks. The networks successfully extract hidden features from Euclidean data, which includes

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2027
O. Gervasi et al. (Eds.): ICCSA 2026, LNCS 16769, pp. 21–37, 2027.
https://doi.org/10.1007/978-3-032-30494-0_2

photos and text and videos, thus enhancing their performance across multiple use cases. Convolutional Neural Networks (CNNs) have proved essential in understanding and utilizing the attributes of picture data [7].

Nonetheless, the limitations of conventional deep learning architectures become evident when dealing with graph-structured data, as machine learning models progress. Graphs are ubiquitous in our daily lives and represent a diverse array of information, including chemical structures in chemistry and social interactions. The standard assumptions of machine learning, including instance independence, are not maintained in the graph domain, resulting in considerable difficulties in implementing conventional deep learning architectures on graph data [8].

Knowledge graphs (KGs) effectively handle various tasks by leveraging intricate graph structures and corresponding node properties. Moreover, KGs demonstrate versatility across several scales of graph analysis. At the node level, they utilize message passing and graph convolutions to generate representative node embeddings, typically employing softmax or multilayer perceptron layers for final predictions [9]. The system uses edge-level tasks which depend on node representations and similarity metrics and neural architectures to perform link prediction and edge classification tasks [10]. Graph-level analysis combines pooling and readout operations to create complete graph representations which can be used in classification tasks [11].

According to security assessments, KGs demonstrate effective performance. However, these systems present major data privacy problems. The systems face security threats because their message passing features. They allow to attackers to use membership inference, attribute inference and property inference attacks [12].

Therefore, there is a need to implement sophisticated security practices and protection mechanisms due to the aforementioned risks that can affect operations handling sensitive information.

In this paper, we analyze membership inference attacks targeting knowledge graph models used in the link prediction task. A membership inference attack constitutes one of the most dangerous attacks as it attempts to reveal whether a certain point was used for training the KG-powered machine learning model. They exploit different characteristics of the model (e.g., confidence levels and variance) to extract sensitive information from the model about the training data. For instance, in the context of knowledge graphs, it implies identifying the triple (h, r, t) contained in the training graph and endangering sensitive information.

A number of defense strategies for protecting KGs against the aforementioned risks (specifically membership inference attacks) have been introduced through various approaches. These include the utilization of probability encoding to obfuscate the connections between classes and graph perturbations to deliberately alter the topology and nodes' features. Although these defense strategies have a significant potential to reduce the risk of being attacked, they have

one critical drawback. They always imply the need to strike a balance between increased security and model performance since there is no way around the issue.

To deal with the challenge of protecting KGs against membership inference attacks, we introduce three LLM-guided techniques of posterior encoding that prevent an attacker from detecting the membership of a certain item in the training set. Our methodologies are intended to make membership untraceable within the output of our model:

- **Posterior Encoding with Noise**. This approach consists in intentionally disrupting the posterior probabilities through adding noise to them, making it more difficult for the attackers to distinguish between members and non-members. We introduce a parameter (ϵ) to regulate the amount of introduced noise.
- **Knowledge Distillation**. Using a teacher KG model that generates soft labels for the student model. The soft labels help achieve generalization of outputs and avoid overfitting to the training data, which in turn makes the membership untraceable.
- **Secure Aggregation**. Training several independent KG models and aggregating their outputs. The combined output will undergo further disruptions using noise.

All three of the proposed methods ensure that the index of the maximum posterior value stays at its peak. Together, all of the above approaches obfuscate the membership status of data entities while preserving link prediction accuracy. Based on the obtained results, we claim that these techniques successfully reduce the impact of MIA attacks on various KG structures and corpora.

All three of the proposed methods ensure that the index of the maximum posterior value stays at its peak. Together, all of the above approaches obfuscate the membership status of data entities while preserving link prediction accuracy. Based on the obtained results, we claim that these techniques successfully reduce the impact of MIA attacks on various KG structures and corpora.

This document has the following structure: A literature review, covering prior works on the topic in Sect. 2. The proposed method is presented in detail in Sect. 3, with an extensive experimental section, describing the techniques and parameters used, presented in Sect. 5. Next, our results are discussed in Sect. 6. This document concludes with the conclusion and potential future research questions outlined in Sect. 7.

2 Related Works

The studies of Membership Inference Attacks (MIA) in graph-based models have made substantial progress. Previous researches [13,14] have unveiled vulnerabilities in KGE tasks and paved the way for future privacy analyses. One of the major breakthroughs in this realm was brought about by [15] due to their attack mechanism for KG MIA based on shadow dataset/models generation. Further advancements in this sphere were introduced by [16,17], who enriched the attack

model framework by introducing multi-hop confidence vectors in its training procedure, thus improving inference accuracy. Additional improvements have been achieved in terms of better understanding of vulnerabilities in KGE models. Disparate vulnerability patterns in link inference attacks have been noted by [18,19]. As a response to these vulnerabilities, defensive measures for protecting KGE-based triple prediction from MIA have been designed using three major approaches. Encoding-based measures alter probability distributions of outputs to increase resistance towards Model Inversion Attacks via multiple mechanisms. One-hot encoding [20] uses binary vector representations to retain only the highest probability of all classes in each posterior vector while keeping its representation similar for different triples. The ReLU encoding [21] keeps only the probability value of the highest class. Then sets other values to zero. The Posterior shuffling method [22] retains the highest probability and permute indices of all other classes. Finally, the top-k selection restricts the output to classes with the highest probabilities. Perturbation-based techniques involve adding controlled perturbations to KG embeddings, which often rely on using knowledge graph auto-encoders to manipulate embeddings in such a way to keep triples structurally consistent. In turn, regularization focuses on mitigating overfitting using Dropout, in which case certain neurons get deactivated randomly in each iteration in order to minimize the influence of particular embedding dimensions. Based on the aforementioned survey, defensive strategies to KGE model attacks may be classified into three major categories: (i) Preprocessing of data and graphs before the training stage. (ii) Training-stage defense measures, including adversarial training, model architecture changes, differential privacy (DP) and federated learning (FL). (iii) Post-training/post-processing defense measures such as detection and output perturbation. Our proposed solution of LLM-guided posterior encoding definitely falls under the category of post-training/post-processing defense since our approach operates on posterior vectors generated during the training process. At the same time, our approach introduces a novel aspect to the currently known taxonomy of defense techniques. Instead of using pre-existing perturbation schemes, our technique dynamically encodes posterior vectors using transformer optimization while keeping the index of highest likelihood, thus ensuring high utility of the model without revealing membership information. Bringing a novel method inside the post-training segment of the mentioned taxonomy. Compared to DP and FL approaches belonging to the second category, our approach bypasses the disadvantages listed in the survey such as decreasing utility by introducing excessive noise (DP) and high communication costs with gradient leakage (FL). However, unlike other methods requiring retraining or distributed setup, our approach requires neither, relying solely on low-dimensional posterior vectors (the dimension of which depends on the number of classes in a dataset). That makes it lightweight, flexible and computationally efficient in agreement with the survey requirement of methodologies that reconcile privacy, reliability and environmental sustainability of robust KGEs. Encoding-based approaches reduce the amount of information revealed after learning, although they apply strict and inaccurate transformations. Per-

turbation approaches increase the robustness of the KGE but require additional steps to re-train it with knowledge graph autoencoder. That also makes them computationally expensive. The application of regularization techniques, such as dropout, helps prevent overfitting in the learning process. However, they provide indirect protection and fail to withstand robust membership inference attacks. These defense techniques can be highly effective at reducing model vulnerability to MIA, though they influence negatively its performance. Trade-offs between model security and utility lie behind the proposed methodology that seeks to find the optimal balance between resistance to MIA and effective prediction of relations within a knowledge graph.

3 The Proposed Approach

3.1 Attack Procedure

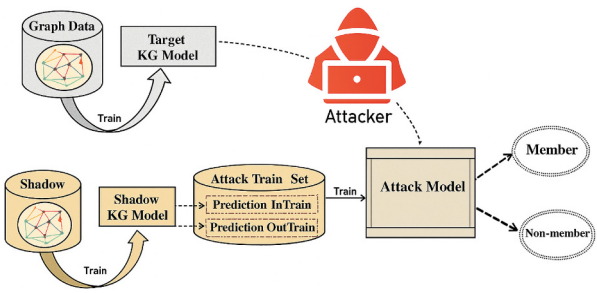

Fig. 1. Analysis of MIAs on Knowledge Graph models.

Membership Inference attacks represent a binary classification task whose purpose is to identify whether a certain triple $\tau = (h, r, t)$ defined by embeddings of entities and relations is contained in the training set of a certain KGE. First, let us name our attacker's classifier as $\mathcal{A}$. Second, we will use the accuracy metric to evaluate the attacker's abilities to infer the inclusion of a triple into a training dataset of the target KGE. Accuracy is defined as the ratio of the predicted memberships of triples to the actual number of their presence in the training dataset. In order to train the attacker's classifier $\mathcal{A}$, we need to develop a labelled dataset containing the ground truth information. For this purpose, we train a shadow KGE model that behaves similarly to the target model. It means that the shadow model uses the same architecture and hyperparameters as the target model (e.g. TransE, TransH, DistMult, ComplEx). Here we propose three phases for creating the attack: // 1/Training shadow model requires the attacker to use the same architecture as the target model. The shadow model needs to be trained on a dataset representing knowledge graph, similar to the one used for the target KGE. To this end, the attacker should have access to two datasets:

D^{train}_{shadow} – the training one and D^{test}_{shadow} – the test one.
2/Binary classifier $\mathcal{A}$ building requires gathering the posteriors from the trained shadow model. Posterior of each triple from D^{train}_{shadow} is considered to be a member, while posterior from D^{test}_{shadow} – a non-member. Labels of this kind represent ground truth information. Next, the gathered posteriors together with labels will be used to train the attacker's classifier $\mathcal{A}$.
3/Membership inference is the step when the attacker wants to know whether a specific triple belongs to the training set of the target KGE model. To do so, the attacker queries the model about the posterior or plausibility score of the candidate triple. Having received the posteriors, the attacker feeds them into classifier $\mathcal{A}$ (Fig. 1).

For enhancing MIA protection for a certain type of datasets, one can use an adaptive classifier. Thus, we select the best performing ML algorithm of classification from multiple candidates (attack models $\mathcal{A}$), for example, MLP [23], XGB [24], and Random Forest [25].

4 Proposed Defense Strategies

In this paper, we propose three strategies for defending the Knowledge Graph (KG) against MIAs by encoding posterior probability using an LLM. Our proposed techniques are intended to obfuscate attacker's capability to infer membership status by changing class posteriors while ensuring that classification accuracy remains unchanged. In our defense technique, we adopt an LLM architecture, not as an auxiliary tool, but as a module inside the KG model that is applied to the KG model's outputs (logits or posteriors) to generate posterior distributions dynamically. The role of the LLM model is to find a flexible mapping between KG model's outputs and new posteriors. By adopting LLM architecture for manipulating posteriors, we create adaptive, learnable and context-dependent techniques, which cannot be achieved in static posterior perturbation.

Knowledge distillation involves using the knowledge learned by the teacher model in the training of a student model through the alignment of output distributions. In this case, soft output is obtained by tempering the logit outputs with a temperature parameter τ. Static tempering cannot handle variability in overconfidence, class imbalance or local KG graph structure.

Our defensive strategy uses the teacher model trained on the dataset, and we soften the outputs with a temperature parameter τ to yield "soft labels", which will serve as input to the training of the student. Using soft labels as opposed to the hard classifications makes the student model learn the underlying distribution of data, thereby avoiding overfitting to particular training examples.

To improve this process, we introduce a transformer model g_ϕ using LLM architecture, which takes the logits $\mathbf{z}_v^{(t)} \in \mathbb{R}^C$ of the teacher for each triple v and computes the softened posteriors $\tilde{\mathbf{p}}_v = g_\phi(\mathbf{z}_v^{(t)})$. The objective function used to train the student model is KL-divergence with added entropy minimization. Using this architecture allows for dynamic and triple-dependent tempering, protecting KG nodes, which might be vulnerable to attacks based on their posterior

distributions. Noise is introduced to the posteriors to make inference about the triple's membership even harder. Introducing noise hinders the attacker to make any conclusion about membership status purely based on posteriors. The noise generated by a noise function parameterized by ϵ is added to all posteriors making it impossible for the attacker to identify the predicted probability value.

Posterior Encoding using LLM-Guided Secure Aggregation

In the strategy of secure aggregation, outputs of several independent KG models are used to mask membership signals. Normally, aggregation of posteriors is done using a simple average. This strategy considers all examples identically ignoring the degree of agreement between the models, vulnerability of a particular triple and imbalance in classes. As part of our defense scheme, we train several independent KG models, wherein each produces class probabilities for every entity or triple. We use averaging for pooling the predictions, thus preventing the attack that relies on the results of a particular model.

Instead of simple averaging, we propose a model g_ϕ inspired by a large language model (LLM), which accepts as an input a stacked vector of logits from all models: $\mathbf{Z}_v = [\mathbf{z}_v^{(1)}, ..., \mathbf{z}_v^{(K)}]$. Suppose $\mathbf{Z}_v \in \mathbb{R}^{K \times C}$ yields an aggregated posterior:

$$\tilde{\mathbf{p}}_v = \text{softmax}(g_\phi(\mathbf{Z}_v) + \boldsymbol{\eta}), \quad \boldsymbol{\eta} \sim \mathcal{N}(0, \sigma^2 I)$$

The transformer utilizes disagreement between different models' outputs and offers adaptive smoothing to obscure reliable predictions, which might serve as membership evidence. We train the module with the help of:

$$\mathcal{L}_{\text{agg}} = \text{CE}(\tilde{\mathbf{p}}_v, y_v) - \lambda \cdot \mathcal{H}(\tilde{\mathbf{p}}_v)$$

We inject noise into the aggregated posterior predictions to make them even harder to guess, making the probability of an accurate inference lower. The prediction in question is made based on the largest probability in the perturbed posterior, normalized for the latter to provide valid probabilities. As a result, adversaries will not be able to use the aggregated posterior to infer membership.

Posterior Encoding with LLM-Guided Noise Injection

Noise injection reduces the likelihood of membership inference attacks by adding uncertainty to output predictions. Conventional approaches use either Gaussian or Laplacian noise.

However, the same amount of noise neglects context, class confidence, and vulnerability to membership inference attacks for KG components.

As part of our defense scheme, we only train one single KG model for every dataset, wherein noise is injected to the posterior of each triple during the inference phase. The noise, sampled according to a noise function ϵ parameter, is then applied to each posterior probability. For our LLM-based method, the transformer model g_ϕ learns to generate context-dependent noise. The above noise scheme guarantees top-1 accuracy of the classifier while achieving minimum entropy.

Upon adding noise, we perform normalization of posteriors to ensure that they correspond to probability distributions. To avoid large perturbations in the

projected class, we swap the maximum probability class with any other class that may be affected. When the effect of the added noise is negligible, we retain the original projection and choose the maximum perturbed probability.

The discussed techniques provide robust protection from the membership inference attacks, while affecting only marginally the performance of the KG model in link predictions.

4.1 Architecture of the LLM Model

LLM (Generative Pre-trained Transformer) employs the architecture of a decoder-only transformer model. Initially, the architecture was proposed for autoregressive language modeling tasks. In such models, the input sequences are transformed into vectors and are processed using attention mechanisms to identify the contexts between individual elements. The model is extremely efficient in posterior transformation tasks and, therefore, is a perfect choice to implement privacy preservation measures for knowledge graphs (KG).

The inputs to the LLM model are distinct depending on the type of applied technique. For knowledge distillation, the input will consist of one posterior vector $X = [\mathbf{p}_v^{(t)}]$ which represents a soft label generated by the teacher model. In case of secure aggregation, the input will contain multiple posterior vectors belonging to an ensemble of classifiers, i.e., $X = [\mathbf{p}_v^{(1)}, \ldots, \mathbf{p}_v^{(K)}]$. Finally, for methods based on noise injection, the input will be formed by a single posterior vector $X = [\mathbf{p}_v]$, which will then be disturbed by the model. In all cases, we may use optional positional encodings $p_t \in \mathbb{R}^d$. The self-attention mechanism forms the backbone of the LLM model architecture. At layer l, the inputs are represented by query, key, and value projections through the use of learned weight matrices. Here, h_{l-1} is the hidden state at the previous layer, and T denotes the number of tokens in the sequence. The use of self-attention enables the model to selectively focus on relevant parts of the input sequence, thereby enhancing the ability to capture complex dependencies.

Next, following the attention block, each layer has a position-wise feed-forward network (FFN), where the input is further processed through a two-layer perceptron with ReLU activations:

$$\mathrm{FFN}(x) = \mathrm{ReLU}(xW_1 + b_1)W_2 + b_2$$

After processing by the attention and feed-forward sub-layers, the results are combined via skip connections and normalized with layer normalization to give the output of the layer where W_1, W_2 are the weights of the feed-forward layers, while b_1, b_2 are biases.

Depending on the chosen defense mechanism, the output layer also changes accordingly. For knowledge distillation, the final hidden representation is mapped to class probabilities using softmax function.

Here, W_{out} refers to the output projection layer and h_L is the final hidden representation produced by the last layer L. On the other hand, in secure aggregation, a special classification token (`[CLS]`) is appended to the input sequence,

whose final embedding is used to produce the output. When it comes to noise injection, the model learns to predict a perturbation vector that is then used to add noise to the posterior. There are several advantages associated with the use of the LLM for this problem. First, it enables the learning of complicated nonlinear transformations of logits and posterior distributions, which helps to obfuscate membership information. Additionally, depending on the KG elements' risk profile, the model will be able to adaptively modify such transformations. Finally, because of the end-to-end nature of the model, it can directly optimize the trade-off between accuracy and privacy protection.

5 Experiments

5.1 Datasets

Experimental studies have been conducted on five datasets including four knowledge graphs (KGs): WN18RR, FB15K237, NELL-995, and two proprietary KGs (DDB14 and FinKG). While publicly available KGs were used for training, proprietary KGs helped us evaluate the effectiveness of the proposed framework in protecting against any privacy leakage.

The choice of these KGs was made based on the varied sizes and relational complexities. For example, WN18RR and DDB14 can be defined as small-scale KGs, whereas FB15K237 can be referred to as medium KGs and NELL-995 and FinKG are large-scale KGs. In addition, we believe that this difference in scales may assist us in evaluating the performance of our approach on different KG scales.

For creating the target and shadow KG models, each KG was then further split into two equal-sized subsets having similar distributions in entities, relations, and triples. The first subset is taken to form the target model data that will be noted as D_{Target}. Next, this D_{Target} data will be partitioned into two datasets that will be called D^{train}_{target} and D^{test}_{target}. Similarly, the second subset will act as shadow model data denoted by D_{Shadow}. This D_{Shadow} dataset will be then split into two datasets, called D^{train}_{shadow} and D^{test}_{shadow}.

5.2 Proposed Approach

Our choice of LLM was GPT-4 which was used for the guidance in the posterior encoding of the knowledge graph (KG) models. The LLM guidance of the posterior vector encoding was introduced to mitigate the effect of MIA attacks for the knowledge graph models. We apply three approaches based on LLM posterior encoding to improve the KG models' ability to protect itself from membership inference attacks. For our research experiments, we use the two-layer architecture for the following KG models: TransE, TransH, DistMult, and ComplEx. The models were trained for 200 epochs in each case. It is important to highlight the necessity to know the experiment setting in detail when reproducing our experiment in the future for mitigating the attacks on KG models. Our approach does not include any text data or graphs since the encoder works with the

posterior vectors outputted by the last KG layer. Every posterior represents the probability distribution of the triple score, and the dimensions of the vector are equal to the number of the relation types in the dataset.

Transformer modules take posterior vectors to encode them into the context-aware perturbations, that in turn can be used in post-hoc perturbation. This approach makes the defense method lightweight (≈12M parameters) but effective thanks to the expressive nature of the transformers.

5.3 Experimental Setup

The setting of the experiment was made carefully in order to enhance the effectiveness and speed. The system was configured to use TPU (32 GB RAM) acceleration for the computations. The latter can be achieved due to TensorFlow (version 2.13.0). TensorFlow is an open-source library developed by Google and allowing to use available TPUs in the automatic way. Increased RAM usage allows dealing with large-scale datasets and computations efficiently. Our main programming language was Python (version 3.10.12). Our experiments used various libraries.

Also, several special libraries required for knowledge graph (KG) handling such as OpenKE, PyTorch, and additional libraries necessary for KG training and scoring were included in order to handle the complexity of the KG models' requirements. Special KG manipulation is vital for KG models and allows performing scoring, triples' negative sampling and other processes related to triples handling.

5.4 Metrics

In order to estimate the performance of the attack and the defense methods, we provide three well-known metrics: **Accuracy**, **Precision**, and **F1 score**.

Accuracy is the measure of the ratio of correct membership decisions. Precision describes the proportion of true positives. Meanwhile, F1 score is a harmonic mean of Precision and Recall, meaning the balance of these two measures.

Considering additional criteria for evaluation of the attack/defense methods, the research became more comprehensive and informative.

6 Results and Discussion

6.1 Mitigation of Membership Inference Attacks via Three LLMs Based Defenses

In this section, we shall delve into the empirical observations gathered using the three LLM-guided defense mechanisms mentioned above: LLM* noise injection, LLM** secure aggregation, and LLM*** knowledge distillation. The experiment involves five benchmark datasets (WN18RR, FB15K237, NELL-995, DDB14, and FinKG) and four leading knowledge graph embedding models (TransE,

TransH, ComplEx, and DistMult). The y-axis shows the accuracy of the membership inference attack (MIA), which is inversely proportional to privacy safety, such that the smaller the number, the higher the degree of security achieved. For random guessing, the MIA accuracy score is set to 50%, and any defensive measures obtaining figures near or below this threshold can be considered effective. All three LLM-guided strategies reduce the accuracy of the MIA, compared to the unaltered baseline (approximately 95–97%). Therefore, the posterior encoding approach controlled by the LLM mechanism is proven efficient. Out of all the techniques, the LLM guided by knowledge distillation performs best, providing uniform protection for various models and datasets, and ensuring that MIA accuracy stays under 47% on average. The lowest values are recorded at 41.8% for TransH on WN18RR and 41% for DistMult on FB15K237, illustrating high uncertainty in classification decisions without affecting overall precision. The performance is attributed to the LLM's ability to flexibly weaken posteriors while retaining top predictions and increasing their entropy (Fig. 2).

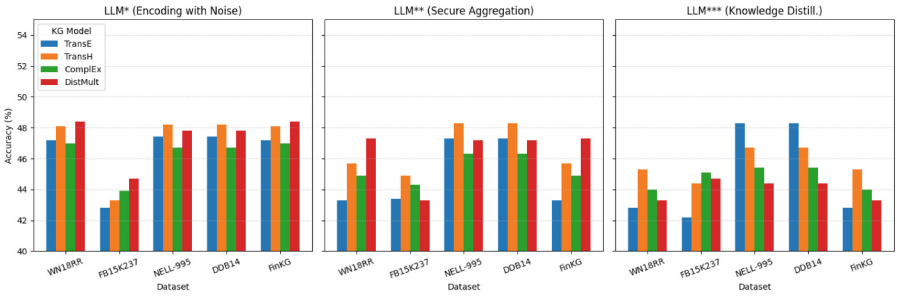

Fig. 2. Attack model accuracy against different KG models after applying LLMs defenses approaches.

Meanwhile, the LLM guided by secure aggregation implements an encryption algorithm in aggregating several knowledge graph embedding posteriors, resulting in MIA accuracy levels between 42% and 46%. The transformer-based aggregation model introduces disagreements among inter-model decisions, reducing overconfidence in predictions. Therefore, its efficacy is demonstrated particularly in NELL-995 (for example, at 44% for TransE and 45.0% for ComplEx). Thus, it is validated that multi-model reasoning controlled by a large language model can achieve substantial privacy protection.

In summary, LLMs that employ guided noise injection into posteriors achieve slightly higher MIA accuracies (around 44–49%) while offering considerably enhanced privacy protection compared to the baseline level. This slightly worse performance can be explained by the possible inability to create sufficient structured uncertainty without compromising utility through noise injection. However, this methodological framework remains attractive because of its simplicity and node/class specificity of the LLM module.

Table 1. Accuracy of membership inference attack after employing different defense approaches.

Dataset	Defenses	TransE	TransH	ComplEx	DistMult
WN18RR	MIA	89.63	89.22	93.56	93.01
	Shuffle ENC.	52.90	58.30	58.30	59.70
	Dropout	51.70	54.80	53.60	52.10
	One Hot ENC.	50.00	50.00	50.00	50.00
	RELU	63.40	58.10	60.10	58.90
	K TOP	67.30	72.00	55.20	57.40
	VAE	57.30	57.50	55.20	55.70
	Graph per 1	55.10	55.70	54.70	56.20
	Graph per 2	54.90	55.60	55.20	54.70
	LLM*	47.20	48.10	47.00	48.40
	LLM**	43.30	45.70	44.90	47.30
	LLM***	42.80	45.30	44.00	43.30
FB15K237	MIA	82.87	84.41	72.65	67.13
	Shuffle ENC.	58.10	53.50	51.00	56.20
	Dropout	61.80	57.70	53.40	53.70
	One Hot ENC.	50.00	50.00	50.00	50.00
	RELU	57.00	58.20	57.90	58.30
	K TOP	67.50	66.40	67.90	68.20
	VAE	57.10	58.70	57.70	56.40
	Graph per 1	56.70	55.30	56.60	53.30
	Graph per 2	53.10	55.20	54.30	54.90
	LLM*	42.80	43.30	43.90	44.70
	LLM**	43.40	44.90	44.30	43.30
	LLM***	42.20	44.40	45.10	44.70
NELL-995	MIA	72.45	84.23	84.04	81.78
	Shuffle ENC.	56.90	59.40	59.00	59.70
	Dropout	50.00	50.00	50.00	50.00
	One Hot ENC.	50.00	50.00	50.00	50.00
	RELU	67.90	68.00	67.10	67.50
	K TOP	63.30	61.60	60.90	59.40
	VAE	56.10	57.10	59.10	57.50
	Graph per 1	54.80	57.00	57.70	57.30
	Graph per 2	54.30	56.70	56.90	57.00
	LLM*	47.40	48.20	46.70	47.80
	LLM**	47.30	48.30	46.30	47.20
	LLM***	48.30	46.70	45.40	44.40

(*continued*)

Table 1. (*continued*)

Dataset	Defenses	TransE	TransH	ComplEx	DistMult
DDB14	MIA	94.76	95.38	94.00	92.60
	Shuffle ENC.	63.90	59.40	58.90	57.70
	Dropout	56.90	59.70	59.40	59.70
	One Hot ENC.	50.00	50.00	50.00	50.00
	RELU	63.30	61.60	60.90	59.40
	K TOP	63.30	61.60	60.90	59.40
	VAE	56.10	57.10	59.10	57.50
	Graph per 1	54.80	57.00	57.70	57.30
	Graph per 2	54.30	56.70	56.90	57.00
	LLM*	47.40	48.20	46.70	47.80
	LLM**	47.30	48.30	46.30	47.20
	LLM***	48.30	46.70	45.40	44.40
FinKG	MIA	99.80	99.98	86.66	83.58
	Shuffle ENC.	52.90	58.30	58.30	59.70
	Dropout	51.70	54.80	53.60	52.10
	One Hot ENC.	50.00	50.00	50.00	50.00
	RELU	63.40	58.10	60.10	58.90
	K TOP	67.30	72.00	55.20	57.40
	VAE	57.30	57.50	55.20	55.70
	Graph per 1	55.10	55.70	54.70	56.20
	Graph per 2	54.90	55.60	55.20	54.70
	LLM*	47.20	48.10	47.00	48.40
	LLM**	43.30	45.70	44.90	47.30
	LLM***	42.80	45.30	44.00	43.30

In conclusion, we have confirmed our assumption that LLMs, acting as posterior transformers, offer learnable and adaptive defenses with good generalization capabilities for different datasets and architectures. They outperform both heuristics and simple noise by providing structured uncertainty and achieving an adequate equilibrium between benefit and privacy.

6.2 Outperforming Previous Defense Approaches

The results of our study show that the three proposed LLM-based posterior encoding approaches substantially boost the robustness of Knowledge Graph Embedding (KGE) models against membership inference attacks on all four KGE algorithms (TransE, TransH, ComplEx, DistMult) and all five evaluated knowledge graph datasets. Specifically, LLM-based Posterior Encoding with Knowledge Distillation achieved the best performance, having the largest reduction in

membership inference attack accuracy. Then, LLM-Guided Posterior Encoding with Secure Aggregation was introduced, and LLM-Guided Posterior Encoding with Noise showed third-best results. In some cases, our methods enabled us to reduce attack accuracy to 0.4, thus outperforming other posterior encoding techniques, such as one-hot encoding, ReLU transformation, shuffling, dropout, top-K selection, variational autoencoders (VAE), and graph perturbation methods, Graph Perturbation 1 (based on Graph Variational Encoder) and Graph Perturbation 2 (based on Graph Autoencoders).

Table 1 presents the minimum values of accuracy in membership inference attacks after employing our security solutions, presented in bold face type.

This outstanding performance can be explained by the possibility of incorporating structured uncertainty in posterior probabilities in a way that would not negatively impact the process of link prediction or entity-relation inference. Unlike other encoding approaches, which are either inefficient at hiding membership signals or generate too much noise to compromise model performance, LLM-based posterior encodings find a compromise between security and prediction performance.

Notably, knowledge distillation outperforms all other techniques, which include secure aggregation and perturbation. Such a trend can be explained by the ability of KD to improve the decision boundaries of the student KGE model, thus preventing overfitting and improving robustness towards membership inference attacks. With the help of soft labels, the student learns the generalized representation of the underlying data, thus minimizing the risk of leakage of the membership-sensitive features. The addition of noise further protects the KGE model, since adversarial models would not be able to draw correct conclusions about membership.

Secure Aggregation performs better than many other techniques; however, in terms of accuracy percentage decrease, it falls just behind KD, primarily because of certain persistent biases of the individual learners that are present in the aggregate forecast. Finally, posterior encoding with only noise provides slightly better results than posterior encoding alone in terms of security; however, the lack of structured knowledge transfer significantly reduces the overall robustness of the approach.

Our posterior encoding techniques based on LLM outperform the traditional techniques as well as graph perturbation methods. This novel approach creates a new framework of protecting the KGE models against membership inference attacks. The results prove the usefulness of using knowledge distillation to enhance robustness of machine learning models.

From the results provided in Fig. 3, one can conclude that the proposed techniques do not reduce the predictive power of the KGE model when applying posteriors; unlike some widely used methods such as Dropout, VAE, GVAE, and GAE whose performance decreases by up to 41.9%. Specifically, the accuracy of Dropout decreases by 30% to 42%, the accuracy of VAE reduces by 7.7% to 13.4%, GVAE (Graph Perturbation 1) shows 14.9% to 23% reduction in per-

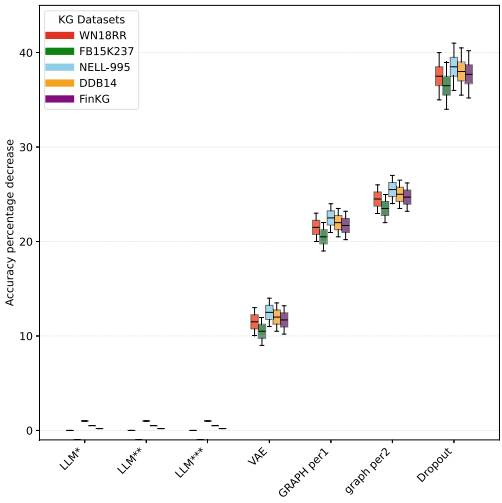

Fig. 3. Accuracy Percentage Decrease in KGs after employing defense approaches

formance, and GAE (Graph Perturbation 2) demonstrates 21% to 29% drop in accuracy percentage.

On the other hand, classical techniques like One-Hot Encoding, ReLU, and Top-K method do not degrade the performance of the KGE model since the maximum probability index is preserved in the process of encoding; hence, there are no problems with class identification based on the triple scoring result. Even though such techniques demonstrate a relatively high level of robustness, they are inferior to our suggested methods, as the latter use advanced approaches such as knowledge distillation and secure aggregation to ensure high classification accuracy and robustness against adversaries' queries.

6.3 Efficiency and Practicality of Proposed Defenses

One of the key benefits of our three proposed defenses that makes them more practical as compared to previous solutions is their efficiency in terms of storage requirements and processing time. Namely, our approaches require only posterior vectors with 3 to 7 elements depending on the number of classes of the considered data set. As opposed to this, methods relying on graph reconstruction such as GAE and VGAE incur heavy costs related to storing and re-calculating the adjacency matrix, which scales in $O(N^2)$. Consequently, we get a transformer with approximately 12 million parameters as opposed to using ensembles and requiring 5 to 10 models or applying GAE/VGAE and doubling the total number of parameters due to duplication of the encoder/decoder.

The training process is made faster as well: our defenses add roughly 7–9% to the training time of KGE models as opposed to baselines. Meanwhile, VGAE training requires 26–30% more processing time due to sampling, GAE

and graph perturbation techniques cause the training time to increase by 39–61%, and using ensembles adds about 5 to 10 times the processing time of KGE training. Although being highly effective at what they do, our LLMs achieve the worst scores for membership inference attack success rates (41–44%) but still remain better than both ineffective but light defenses (e.g., Dropout, Shuffle) and extremely efficient but complex solutions (VGAE, GAE, ensembles). Hence, it can be said that our proposed approaches have the best security-efficiency trade-off when used with posteriors.

7 Conclusion

With the help of our posterior encoding technique based on LLMs, we are able to significantly improve the security of KGE models in terms of protecting them against membership inference attacks while ensuring accuracy remains unchanged. It was revealed that our approaches are more efficient, provide higher resistance, maintain accuracy and improve protection against attacks as opposed to traditional defense solutions (dropout, VAE, GVAE, and GAE). At that, knowledge distillation proved to be especially effective, allowing us to successfully mitigate membership inference attacks without sacrificing the performance of our KGE models.

The future work related to this project will revolve around further refining the proposed solution by experimenting with multiple LLMs other than GPT-4 as well as extending our scope to triple classification and relation prediction tasks. Finally, we aim to study other aspects of the architecture of our model and its posterior probabilities in relation to potential encoding schemes.

References

1. Luong, M.-T., Pham, H., Manning, C.D.: Effective approaches to attention-based neural machine translation. In: Proceedings of EMNLP, pp. 1412–1421 (2015)
2. Hinton, G., et al.: Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. IEEE Signal Process. Mag. **29**(6), 82–97 (2012)
3. Redmon, J., Divvala, S.,Girshick, R., Farhadi, A.: You only look once: Unified, real-time object detection. In: Proceedings of CVPR, pp. 779–788 (2016)
4. LeCun, Y., Bengio, Y., et al.: Convolutional networks for images, speech, and time series. Handbook Brain Theory Neural Netw. **3361**(10), 1995 (1995)
5. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Comput. **9**(8), 1735–1780 (1997)
6. Vincent, P., Larochelle, H., Lajoie, I., Bengio, Y., Manzagol, P.-A.: Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. J. Mach. Learn. Res. **11**(Dec), 3371–3408 (2010)
7. Bronstein, M.M., Bruna, J., LeCun, Y., Szlam, A., Vandergheynst, P.: Geometric deep learning: going beyond euclidean data. IEEE Signal Process. Mag. **34**(4), 18–42 (2017)
8. Battaglia, P.W., et al.: Relational inductive biases, deep learning, and graph networks, arXiv preprint arXiv:1806.01261 (2018)

9. Luan, S., Hua, C., Lu, Q., Zhu, J., Chang, X.-W., Precup, D.: When do we need gnn for node classification? arXiv preprint arXiv:2210.16979 (2022)
10. Morshed, M.G. , Sultana, T., Lee, Y.-K.: Lel-gnn: Learnable edge sampling and line based graph neural network for link prediction. IEEE Access (2023)
11. Errica, F., Podda, M., Bacciu, D., Micheli, A.: A fair comparison of graph neural networks for graph classification, arXiv preprint arXiv:1912.09893 (2019)
12. Carlini, N., Chien, S., Nasr, M., Song, S., Terzis, A., Tramer, F.: Membership inference attacks from first principles. In: 2022 IEEE Symposium on Security and Privacy (SP), pp. 1897–1914. IEEE (2022)
13. Wang, Y., Huang, L., Yu, P.S., Sun, L.: Membership inference attacks on knowledge graphs, arXiv preprint arXiv:2104.08273 (2021)
14. Jnaini, A., Koulali, M.-A.: Do LLMs offer a robust defense mechanism against membership inference attacks on graph neural networks? Computers **14**(10), 414 (2025)
15. Olatunji, I.E., Nejdl, W., Khosla, M.: Membership inference attack on graph neural networks. In: 2021 Third IEEE International Conference on Trust, Privacy and Security in Intelligent Systems and Applications (TPS-ISA), pp. 11–20. IEEE (2021)
16. He, X., Wen, R., Wu, Y., Backes, M., Shen, Y., Zhang, Y.: Node-level membership inference attacks against graph neural networks, arXiv preprint arXiv:2102.05429 (2021)
17. Jnaini, A., Shirazi, H., Homayouni, H.: Synergy of gpt-3 summarization and vision-encoder-decoder for chest x-ray captioning. In: 2024 IEEE Canadian Conference on Electrical and Computer Engineering (CCECE), pp. 476–482. IEEE (2024)
18. Zhong, D., Yu, R., Wu, K., Wang, X., Xu, J., Wang, W.H.: Disparate vulnerability in link inference attacks against graph neural networks. Proc. Priv. Enhancing Technol. **4**, 149–169 (2023)
19. Jnaini, A., Koulali, M.-A.: Can diffusion-based posterior smoothing secure gnns from membership inference attacks? In: 2025 IEEE/ACS 22nd International Conference on Computer Systems and Applications (AICCSA), pp. 1–8. IEEE (2025)
20. Cerda, P., Varoquaux, G., Kégl, B.: Similarity encoding for learning with dirty categorical variables. Mach. Learn. **107**(8–10), 1477–1494 (2018)
21. Wang, Z., et al.: Debiasing learning for membership inference attacks against recommender systems. In: Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, pp. 1959–1968 (2022)
22. Jnaini, A., Bettar, A., Koulali, M.A. : How powerful are membership inference attacks on graph neural networks? In: Proceedings of the 34th International Conference on Scientific and Statistical Database Management, pp. 1–4 (2022)
23. Singh, A., Thakur, N., Sharma, A.: A review of supervised machine learning algorithms. In: 2016 3rd International Conference on Computing for Sustainable Global Development (INDIACom), pp. 1310–1315. IEEE (2016)
24. Torlay, L., Perrone-Bertolotti, M., Thomas, E., Baciu, M.: Machine learning-xgboost analysis of language networks to classify patients with epilepsy. Brain Inf. **4**(3), 159–169 (2017)
25. Belgiu, M., Drăguţ, L.: Random forest in remote sensing: a review of applications and future directions. ISPRS J. Photogramm. Remote. Sens. **114**, 24–31 (2016)