A Framework for Differentially Private Knowledge Graph Embeddings

Khushi Pujara and Disha Suthar

Prof. Sayak Ray Chowdhury
[CS798L]

Department of Computer Science, IIT Kanpur

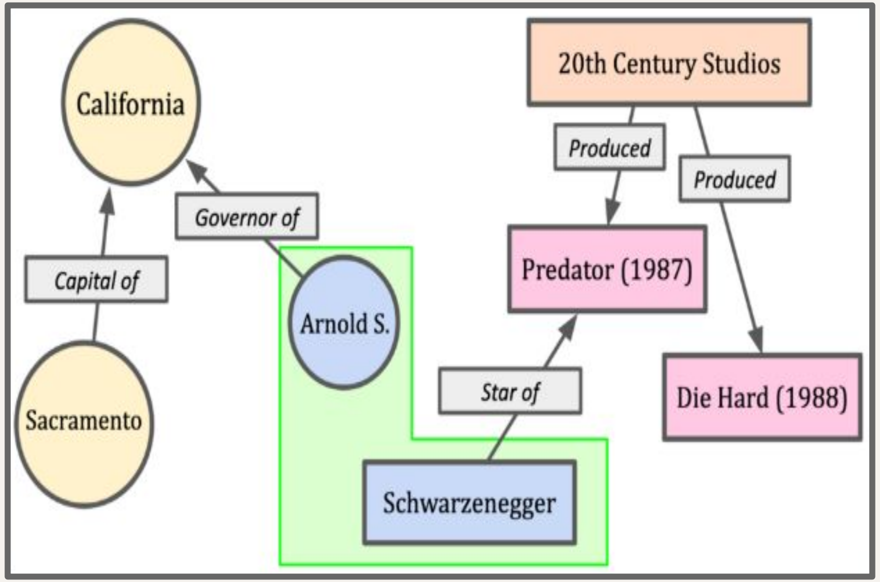
01
Understanding the Paper

Knowledge Graphs (KGs)

A knowledge graph is like a network of facts, where:

- Nodes represent entities (like people, places, things)
- Edges represent relationships between those entities

Formally, Let E and L denote the set of entities and relationships. A knowledge graph $K \subset E \times L \times E$ is a set of statements (h, l, t), where $h, t \in E$ and $l \in L$.



The Privacy Problem in Knowledge Graphs

- Knowledge Graphs store sensitive information.
- Organizations have data they want to share but can't due to privacy concerns
- Example : an NGO like HINCare, which helps elderly people, wants to share data but can’t because of privacy laws. They risk legal trouble if they accidentally reveal personal information.

💡 What’s the New Idea?

Instead of sharing the raw KG data (which might include private info), we can convert it into embeddings — basically, vectors (numbers) that capture the structure of the KG. These are great for tasks like:

- Finding missing links
- Grouping similar entities
- Answering questions

But even these embeddings can still leak private info.

Differentially Private Knowledge Graph Embeddings

- This paper introduces a new framework called DPKGE.
  It uses a privacy technique called Differentially Private Stochastic Gradient Descent (DP-SGD) during training. This means it adds noise while learning from the data, to hide sensitive facts.

- It builds on existing methods like TransE (standard KG embedding techniques) and makes them private.

- It shows that only protecting the sensitive parts of the KG gives better results — privacy and usefulness can coexist.

KG Embedding Methods

1. Translational Methods

They map entities and relations into the same vector space and treat relations like translations.
Example:

- TransE: Models relation l as a translation from h to t. So, h + l ≈ t. Variants like:
- TransH: Translation on a hyperplane.
- TransR: Entities and relations in separate spaces.
- TransD: Better handles graph diversity with fewer parameters.

2. Bilinear Methods

Use quadratic (bilinear) functions to model interactions.
Examples:

- RESCAL: Tensor factorization.
- DistMult: Simplifies computation by restricting relation matrices.
- ComplEx: Uses complex numbers to model symmetric/asymmetric relations.
- TuckER: Uses Tucker decomposition for efficient embedding.

Differential Privacy for Graphs

Neighboring Graphs:

- Two graphs are edge-neighbors if they differ by one edge.
- Node-neighbors differ by a node and its edges.

Why Edge-level DP for KGs?

- KGs are multi-relational and directed.
- If you represent a node as a vector (embedding), then its presence can be detected, breaking node-level DP.
- Hence, this paper uses edge-DP, which is more practical.

Effect of Privacy

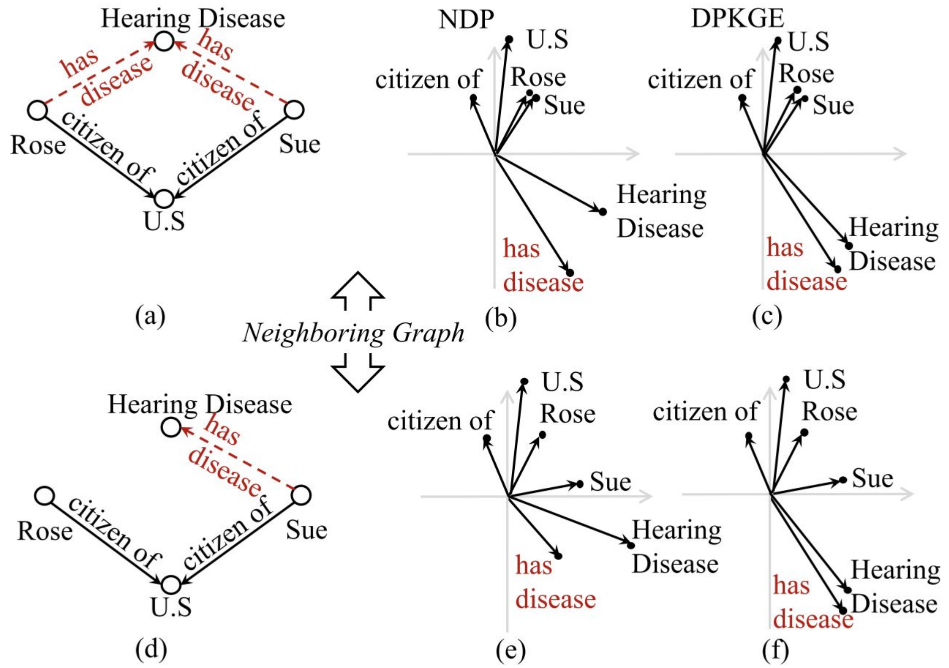


Fig. 1. A graph with unrestricted statements (solid black line) and confidential statements (dashed red line). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

DPKGE: Differentially Private KG Embedding

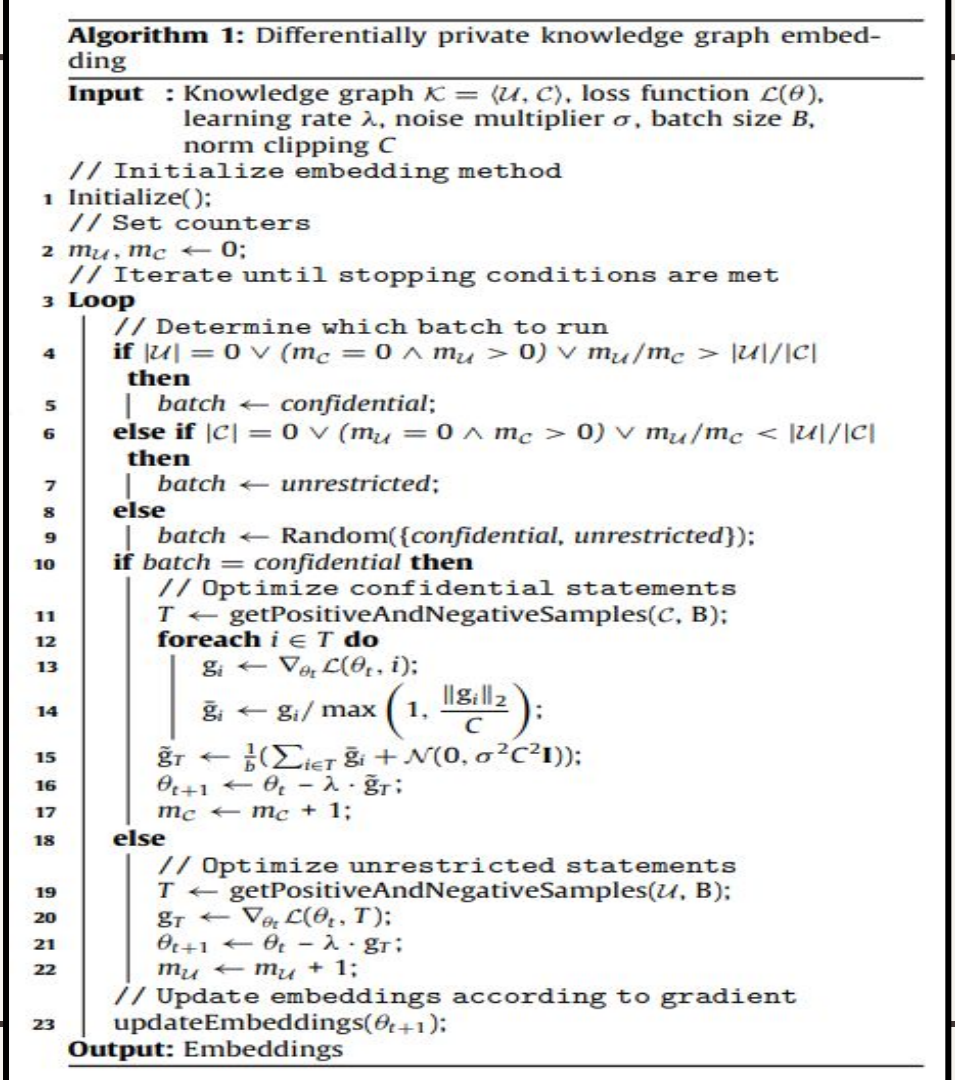
Objective: Adapt existing embedding methods to ensure differential privacy.

Key Features:

- Applies differential privacy only to confidential statements
- Achieves a balance between training on public and confidential statements
- Preserves the structural integrity of the learned embeddings

Requirements:

- The input knowledge graph must use a gradient-separable embedding algorithm
- Embedding updates should be performed via gradient descent or other gradient-based optimization methods



Algorithm 1: Differentially private knowledge graph embedding

```
Input  : Knowledge graph $\mathcal{K} = \langle \mathcal{U}, \mathcal{C} \rangle$, loss function $\mathcal{L}(\theta)$,
         learning rate $\lambda$, noise multiplier $\sigma$, batch size $B$,
         norm clipping $C$
// Initialize embedding method
Initialize();
// Set counters
$m_{\mathcal{U}}, m_{\mathcal{C}} \leftarrow 0$;
// Iterate until stopping conditions are met
Loop
    // Determine which batch to run
    if $|\mathcal{U}| = 0 \vee (m_{\mathcal{C}} = 0 \wedge m_{\mathcal{U}} > 0) \vee m_{\mathcal{U}}/m_{\mathcal{C}} > |\mathcal{U}|/|\mathcal{C}|$
     then
        $batch \leftarrow confidential$;
    else if $|\mathcal{C}| = 0 \vee (m_{\mathcal{U}} = 0 \wedge m_{\mathcal{C}} > 0) \vee m_{\mathcal{U}}/m_{\mathcal{C}} < |\mathcal{U}|/|\mathcal{C}|$
     then
        $batch \leftarrow unrestricted$;
    else
        $batch \leftarrow$ Random({$confidential$, $unrestricted$});
    if $batch = confidential$ then
        // Optimize confidential statements
        $T \leftarrow$ getPositiveAndNegativeSamples($\mathcal{C}$, B);
        foreach $i \in T$ do
            $g_i \leftarrow \nabla_{\theta_t} \mathcal{L}(\theta_t, i)$;
            $\bar{g}_i \leftarrow g_i / \max\left(1, \frac{\|g_i\|_2}{C}\right)$;
        $\tilde{g}_T \leftarrow \frac{1}{b}(\sum_{i \in T} \bar{g}_i + \mathcal{N}(0, \sigma^2 C^2 \mathbf{I}))$;
        $\theta_{t+1} \leftarrow \theta_t - \lambda \cdot \tilde{g}_T$;
        $m_{\mathcal{C}} \leftarrow m_{\mathcal{C}} + 1$;
    else
        // Optimize unrestricted statements
        $T \leftarrow$ getPositiveAndNegativeSamples($\mathcal{U}$, B);
        $g_T \leftarrow \nabla_{\theta_t} \mathcal{L}(\theta_t, T)$;
        $\theta_{t+1} \leftarrow \theta_t - \lambda \cdot g_T$;
        $m_{\mathcal{U}} \leftarrow m_{\mathcal{U}} + 1$;
    // Update embeddings according to gradient
    updateEmbeddings($\theta_{t+1}$);
Output: Embeddings
```

The Algorithm

Experimental Setup

Datasets:

- Standard KG datasets: FB15k, FB15k-237, YAGO3-10 (Randomly set percentage of statements as confidential)
- Health datasets with natural confidential statements: MIMIC-III, eICU

Methods:

- DPKGE: TransE_DPKGE, TransM_DPKGE, RESCAL_DPKGE, DistMult_DPKGE
- Non-DP (NDP): Standard embedding methods
- FullDP: DP on all statements (confidential and unrestricted)

Evaluation metrics: Mean Rank (MR) and Hits@10 (Hits)

Key Experimental Results

Loss Convergence

- DP-KGE methods, which apply differential privacy only to confidential statements, demonstrate good convergence behavior.
- Their loss values are closer to those of non-DP (NDP) methods compared to FullDP methods.
- In contrast, FullDP methods with high noise levels (e.g., $\sigma = 10$) exhibit difficulty in convergence.

Key Experimental Results (Contd.)

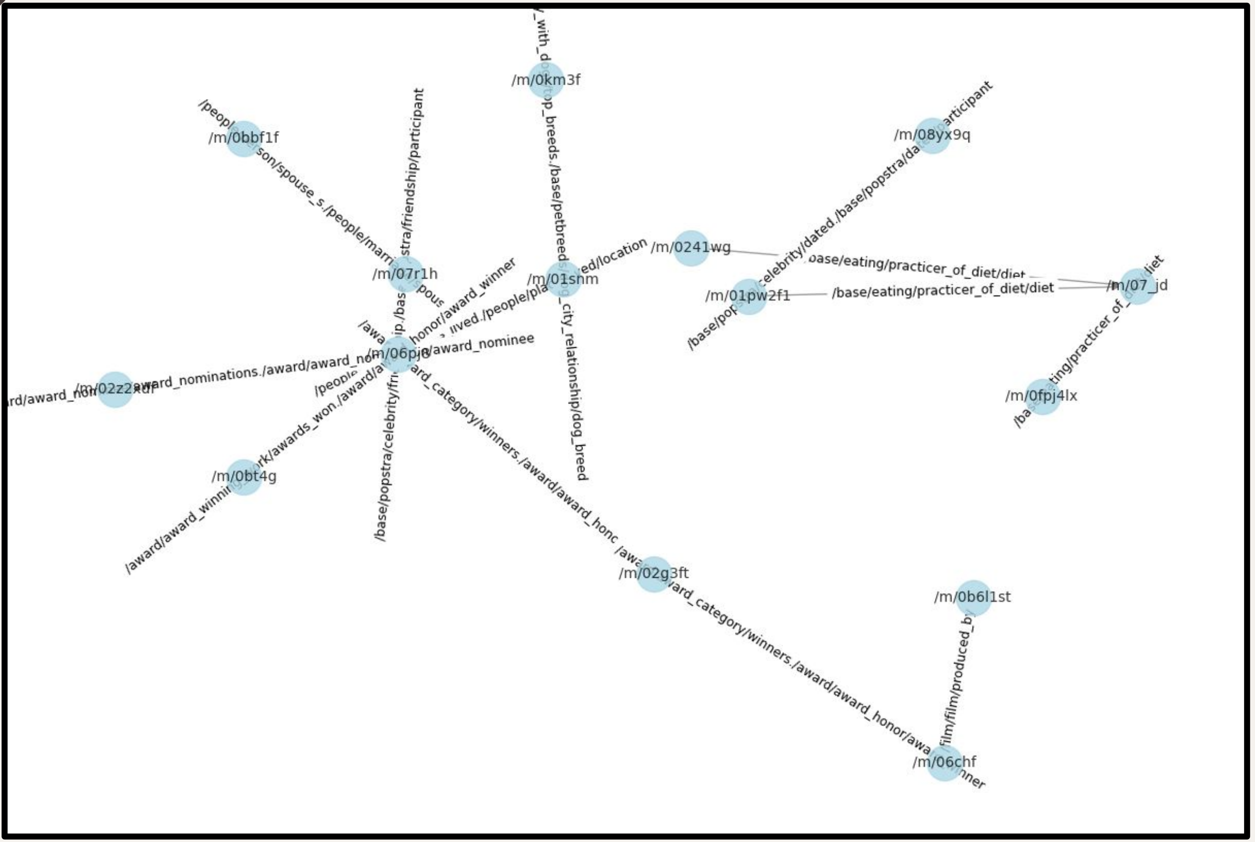
Privacy Budget Behavior

- The privacy budget (ε) tends to grow linearly after approximately 10 training epochs.
- A lower noise level (σ) results in a steeper increase in ε over time.
- The size of the dataset does not significantly influence the pattern of ε growth.

02
Our Approach

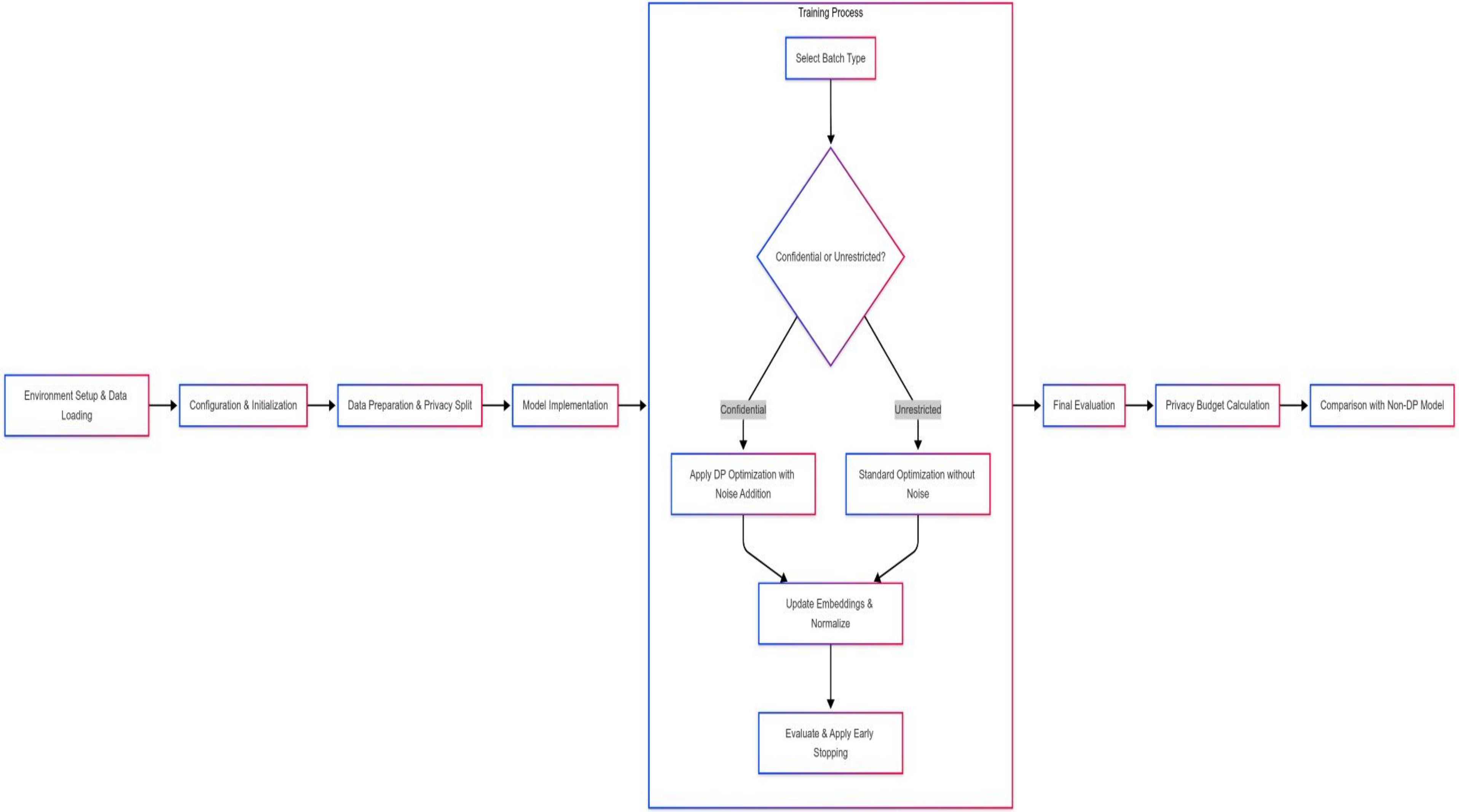
Setup

- Model type : Translation-based embedding model (TransE).
- Dataset : FB15k-237 (a large knowledge graph dataset).
- Training Objective : Predict missing triples in the knowledge graph (link prediction task).



/m/0km3f
/m/0bbf1f
/m/08yx9q
/m/07r1h
/m/01snm
/m/0241wg
/m/01pw2f1
/m/07_jd
/m/06pj8
/m/02z2xdf
/m/0fpj4lx
/m/0bt4g
/m/02g3ft
/m/0b6l1st
/m/06chf
/base/eating/practicer_of_diet/diet
/base/popstra/celebrity/friendship/participant
/award/award_nominee/award_nominations./award/award_nomination/award_nominee
/award/award_winning_work/awards_won./award/award_honor/award_winner
/award/award_category/winners./award/award_honor/award_winner
/base/petbreeds/city_with_dogs/top_breeds./base/petbreeds/dog_city_relationship/dog_breed
/base/popstra/celebrity/dated./base/popstra/dated/participant
/people/person/spouse_s./people/marriage/spouse
/people/person/places_lived./people/place_lived/location
/film/film/produced_by

Sample Knowledge Graph from fb15k-237 dataset


Environment Setup & Data Loading
Configuration & Initialization
Data Preparation & Privacy Split
Model Implementation
Training Process
Select Batch Type
Confidential or Unrestricted?
Confidential
Unrestricted
Apply DP Optimization with Noise Addition
Standard Optimization without Noise
Update Embeddings & Normalize
Evaluate & Apply Early Stopping
Final Evaluation
Privacy Budget Calculation
Comparison with Non-DP Model

Differential Privacy Integration

Key Privacy Mechanisms:

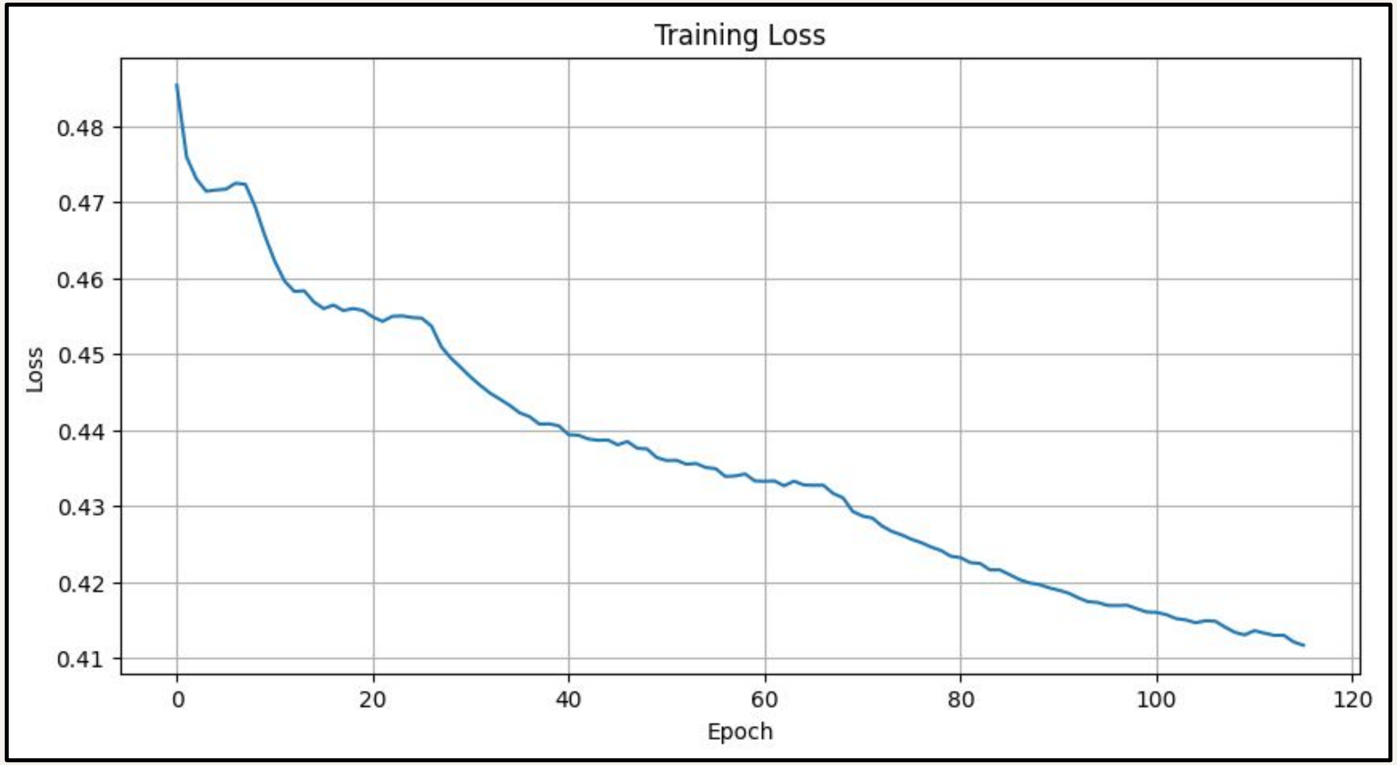
- Noise Addition: Gaussian noise added to gradients for differential privacy.
- Gradient Clipping: Ensures gradients don’t leak too much information.
- Privacy Budget: Set as $\varepsilon = 3.158$ for the experiment.

Helper Functions :

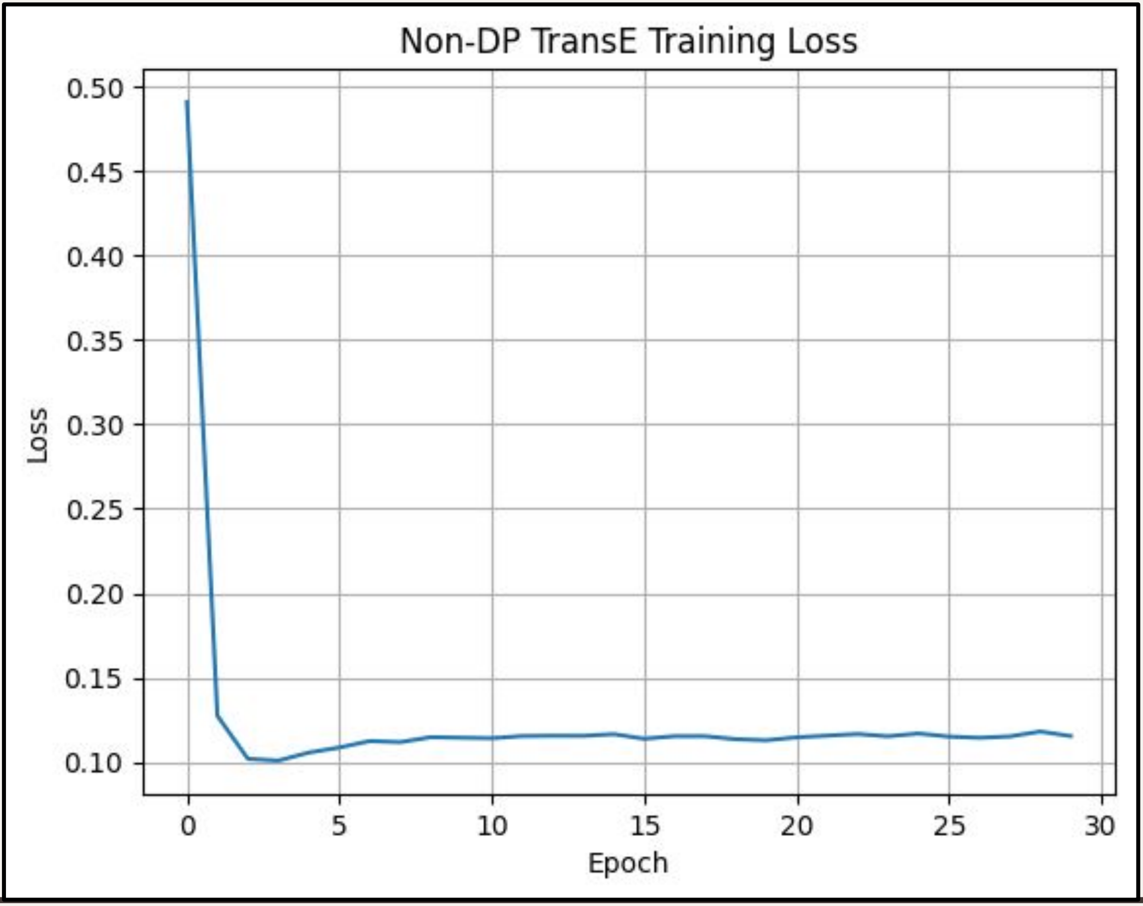
- L2_distance : Computes L2 distance for TransE (i.e., $||h+r-t||_2$|).
- Forward : Performs forward pass for a batch of triples.
- Loss_function : Computes margin-based loss with L2 regularization.
- Optimize_confidential : Optimization step with noise addition for differential privacy.
- Optimize_unrestricted : Regular optimization without privacy mechanisms.

Evaluation Metrics

- Mean Rank (MR): It calculates the average rank of the correct entity when ranking all possible entities for each triple in the test set. A lower MR indicates better performance.
- Mean Reciprocal Rank (MRR): MRR is the average of the reciprocal ranks of the correct entity. It emphasizes the ranking of correct entities in the top positions. A higher MRR is better.
- Hits@K: This measures the proportion of correct entities ranked in the top K positions. It's often used for evaluation in tasks like knowledge graph completion. Higher values of Hits@K indicate better performance, particularly when the correct answer appears in the top K ranks.



Training Loss of DP-KGE



Training Loss of Non-DP KGE

Metrics

- Mean Rank (MR): It calculates the average rank of the correct entity when ranking all possible entities for each triple in the test set. A lower MR indicates better performance.
- Mean Reciprocal Rank (MRR): MRR is the average of the reciprocal ranks of the correct entity. It emphasizes the ranking of correct entities in the top positions. A higher MRR is better.
- Hits@K: This measures the proportion of correct entities ranked in the top K positions. It's often used for evaluation in tasks like knowledge graph completion. Higher values of Hits@K indicate better performance, particularly when the correct answer appears in the top K ranks.

Results

DP-KGE	Non-DP KGE
Hits@10 = 0.154	Hits@10 = 0.185
MR = 4973.18	MR = 687.83
MRR = 0.0639	MRR = 0.089

Insights

- DP causes a performance drop, as expected due to the added noise for privacy.

- MR increased significantly under DP, indicating that the model ranks correct entities much farther down the list.

- MRR and Hits@10 also drop, but not as drastically — which means that in a small subset of cases, the model still ranks the right entity reasonably high.

Future Work

- Implement Full DP and compare results of all three.
- Check for different values of noise multipliers and analyse its effects.
- Try DP optimizers such as DPAdam, DPAdaGrad etc.
- Compute evaluation 5 times and report statistics such as mean and standard deviation of each metrics as well.

THANK YOU!