Towards Forgetting in Knowledge Graphs: A Unified Framework for Machine Unlearning and Differential Privacy

Anonymous ACL submission

Abstract

Knowledge graphs encode rich semantic relationships but raise privacy concerns when containing sensitive information.

Existing machine unlearning methods enable selective data deletion from trained models but lack privacy guarantees during query answering.

Differential privacy mechanisms protect training data but provide no mechanism for post-hoc deletion when users exercise legal rights.

We present a unified framework combining certified machine unlearning with differential privacy, proposing three complementary algorithms (Influence-based Selective Retraining, Sharded Knowledge Graph Learning, Incremental Graph Forgetting), privacy-preserving query mechanisms for SPARQL and link prediction, and an integrated compliance architecture. Experimental validation on synthetic knowledge graphs demonstrates Influence-based Selective Retraining achieves approximation error below 0.1 with deletion completing in under 40 seconds for 10% data removal, enabling practical deployment under regulatory constraints.

1 Introduction

Knowledge graphs have become foundational for modern AI systems, powering search engines, recommendation systems, and drug discovery. However, their widespread adoption creates a fundamental tension: the interconnectedness that makes knowledge graphs powerful also makes them vulnerable to privacy violations.

Consider a practical scenario. A user requests deletion of their data under GDPR Article 17, the "right to be forgotten." An organization can easily delete raw records by removing rows from tables and deleting files from storage. But the knowledge has already been learned by machine learning models trained on that data. The user's preferences, behaviors, and relationships have been encoded into embedding vectors and embedded into model parameters. Simply deleting raw data leaves these learned patterns intact, creating the "forgetting paradox": data can be removed from storage, but knowledge persists indefinitely in models.

This problem is particularly acute for knowledge graphs due to structural dependencies. An entity's representation doesn't exist in isolation; it's shaped by its neighbors, creating cascading effects throughout the graph. Deleting one person's data should remove not just their entity node, but also any influence their data had on learned representations of connected entities.

Current approaches address either machine unlearning or differential privacy in isolation. Machine unlearning methods focus on efficient deletion through influence-based updates or partition-based retraining, but provide no guarantees against privacy leakage during query answering. Differential privacy methods protect training data through noise injection, but lack mechanisms for post-hoc deletion when users exercise legal rights. No existing framework provides both certified unlearning and query-time privacy protection with auditable compliance mechanisms.

We present the first unified framework bridging this gap with five key contributions. First, we provide rigorous problem formalization defining three deletion types (entity, relation, and subgraph deletion) with complexity characterization. Second, we propose three complementary unlearning algorithms optimized for different scenarios: Influence-based Selective Retraining (ISR) for fast approximate unlearning, Sharded Knowledge Graph Learning (SKGL) for exact guarantees, and Incremental Graph Forgetting (IGF) for real-time deletion. Third, we design differentially private

query mechanisms for SPARQL, link prediction, and pattern mining. Fourth, we present end-to-end system architecture with cryptographic compliance certificates. Fifth, we establish fundamental bounds on the privacy-utility-efficiency tradeoff and validate through experiments achieving approximation error below 0.1 with deletion completing in under 40 seconds for 10% data removal.

2 Related Work

Research on privacy-preserving knowledge graphs has evolved through three parallel streams: machine unlearning for trained models, differential privacy for data protection, and knowledge graph embedding methods. Our work synthesizes these directions into a unified framework addressing their complementary limitations.

2.1 Machine Unlearning Foundations

The machine unlearning literature establishes methods for removing training data influence from learned models without complete retraining. Cao and Yang (2015) introduced the foundational concept of making systems forget through selective parameter updates. Bourtoule et al. (2021) proposed SISA (Sharded, Isolated, Sliced, and Aggregated learning), which partitions training data into isolated shards enabling efficient localized retraining upon deletion requests. Their approach demonstrates that strategic data organization during training can dramatically reduce unlearning costs.

Recent work extends unlearning to graph-structured data. Chen et al. (2022) developed the first graph unlearning framework addressing structural dependencies in social networks. Cheng et al. (2022) introduced GNNDelete, providing general strategies for unlearning in graph neural networks through gradient-based approximations. Chien et al. (2022) advanced the field by providing certified guarantees for graph unlearning, establishing theoretical bounds on approximation quality. The OpenGU framework (OpenGU Contributors, 2025) provides open-source implementations demonstrating practical deployability of these techniques.

However, existing graph unlearning research exhibits three critical gaps. First, methods focus exclusively on the unlearning operation itself without addressing privacy leakage through subsequent query answers. Second, most approaches target general social network graphs rather than knowledge graph semantics with typed relations and heterogeneous entity types. Third, none integrate unlearning with differential privacy guarantees for comprehensive data protection. Our framework addresses these limitations through joint unlearning and privacy mechanisms specifically designed for knowledge graph workloads.

2.2 Differential Privacy for Graphs

Differential privacy provides formal guarantees that aggregate query results do not reveal information about individual data points. Dwork et al. (2006) established the foundational Laplace mechanism for numeric queries, while Dwork and Roth (2014) developed comprehensive theoretical foundations spanning composition theorems and privacy accounting methods. Abadi et al. (2016) adapted differential privacy to deep learning through DP-SGD, enabling private training of neural networks via gradient clipping and noise injection.

Extensions to graph data address unique challenges from structural dependencies. Han et al. (2022) developed differentially private knowledge graph embedding methods balancing utility and privacy through adaptive noise calibration. Wei et al. (2024) introduced Poincaré differential privacy for hierarchy-aware graph embeddings, recognizing that hyperbolic geometry better captures hierarchical structures common in knowledge graphs. Chen et al. (2021) addressed federated scenarios where knowledge graphs span multiple organizations, designing protocols that protect both node attributes and graph structure.

Despite these advances, differential privacy methods for graphs lack unlearning capabilities. Once a model is trained with privacy guarantees, there exists no mechanism to selectively remove specific entities or relations when users exercise deletion rights under data protection regulations. Organizations must choose between maintaining privacy guarantees (requiring full retraining) or efficiently handling deletions (sacrificing formal privacy). Our framework resolves this dilemma by designing unlearning algorithms that preserve differential privacy guarantees throughout the deletion lifecycle.

2.3 Knowledge Graph Embeddings

Knowledge graph embedding methods learn continuous vector representations capturing semantic relationships between entities and relations. Bordes et al. (2013) introduced TransE, modeling relations as translations in embedding space. Sun et al. (2019) proposed RotatE, representing relations as rotations in complex vector space to better capture symmetry and antisymmetry patterns. Trouillon et al. (2016) developed ComplEx embeddings using complex-valued representations for enhanced expressiveness.

These embedding methods enable downstream applications including link prediction for knowledge graph completion, entity resolution across heterogeneous sources, and complex query answering. However, standard approaches have no built-in privacy or unlearning mechanisms. When trained on sensitive data, embeddings persistently encode information about deleted entities even after removal from the underlying graph store. Furthermore, embedding-based query answering can leak private information through carefully constructed adversarial queries.

Koh and Liang (2017) introduced influence functions for interpreting predictions by tracing them back to training data. Guo et al. (2020) adapted influence functions for certified data removal, establishing theoretical foundations for approximate unlearning. We build on these techniques while extending them to knowledge graph semantics and integrating them with differential privacy mechanisms.

Recent surveys by Nguyen et al. (2022) and Ji et al. (2014) provide comprehensive overviews of machine unlearning and differential privacy respectively, identifying the gap we address: no existing work provides an integrated framework combining certified unlearning with differential privacy specifically designed for knowledge graph workloads. Our contribution fills this gap through novel algorithms, privacy mechanisms, and compliance architectures that jointly satisfy both unlearning requirements and privacy guarantees.

3 Problem Formalization

We formalize knowledge graph unlearning under differential privacy. A knowledge graph is $\mathcal{G} = (\mathcal{E}, \mathcal{R}, \mathcal{T})$ where $\mathcal{E}$ is the set of entities, $\mathcal{R}$ is the set of relations, and $\mathcal{T} \subseteq \mathcal{E} \times \mathcal{R} \times \mathcal{E}$ is the set of triples. Given embedding function $\phi : \mathcal{E} \cup \mathcal{R} \to \mathbb{R}^d$ and trained model parameters θ, a deletion request specifies data $\mathcal{D} \subseteq \mathcal{T}$ to remove.

The unlearning problem is: given model M_θ trained on $\mathcal{G}$, produce updated model $M_{\theta'}$ that behaves as if trained on $\mathcal{G} \setminus \mathcal{D}$, while being computationally efficient and providing privacy guarantees.

We consider three deletion types with increasing complexity. Entity deletion removes an entity and all incident triples. Relation deletion removes all triples with a specific relation type. Subgraph deletion removes an arbitrary set of triples, the most general and challenging case.

3.1 Privacy Requirements

Definition 1 (Node Differential Privacy)**.** *Mechanism $\mathcal{M}$ is (ϵ, δ)-node DP if for any neighboring knowledge graphs $\mathcal{G}, \mathcal{G}'$ differing by one entity and all its incident edges, and for any subset of outputs $\mathcal{S}$:*

$$P(\mathcal{M}(\mathcal{G}) \in \mathcal{S}) \leq e^\epsilon \cdot P(\mathcal{M}(\mathcal{G}') \in \mathcal{S}) + \delta \quad (1)$$

Definition 2 (Edge Differential Privacy)**.** *(ϵ, δ)-edge DP holds when $\mathcal{G}, \mathcal{G}'$ differ by a single triple (h, r, t).*

Node DP has higher sensitivity due to protecting all incident edges. For power-law graphs with very large maximum degree, this creates severe utility challenges.

3.2 Unlearning Quality Metrics

We measure unlearning quality through approximation error. Let M^* be a model retrained from scratch on $\mathcal{G} \setminus \mathcal{D}$, and M' be the unlearned model. The approximation error is $\epsilon_{approx} = \mathbb{E}[\|M'(x) - M^*(x)\|]$ where the expectation is over test queries x.

We also measure deletion time, affected entity count, and utility preservation through link prediction accuracy on retained data.

4 Certified Unlearning Algorithms

4.1 Influence-based Selective Retraining

ISR leverages influence functions to approximate the effect of deleting training samples, then selectively updates affected parameters. For knowledge graph embeddings, we track how entity and relation embeddings influence each other through the training objective.

For each triple $(h, r, t) \in \mathcal{D}$ to be deleted, we compute the influence on model parameters:

$$I_{(h,r,t)} = \nabla_\theta L(\theta)^\top H^{-1} \nabla_\theta \ell_{(h,r,t)} \tag{2}$$

where $L(\theta)$ is the total loss, $\ell_{(h,r,t)}$ is the loss for triple (h, r, t), and $H = \nabla^2_\theta L(\theta)$ is the Hessian matrix capturing second-order interactions.

We identify affected entities as $\mathcal{E}_{affected} = \{e \in \mathcal{E} : \|I_e\| > \tau\}$ where τ is a threshold parameter controlling the tradeoff between speed and accuracy. Parameters are updated via reverse gradient:

$$\theta' = \theta - \eta \sum_{(h,r,t)\in\mathcal{D}} I_{(h,r,t)} \tag{3}$$

Finally, we refine affected parameters on $\mathcal{G} \setminus \mathcal{D}$ for k epochs using standard gradient descent.

Theorem 1 (ISR Approximation Bound)**.** *If the Hessian H is L-smooth and the loss function is μ-strongly convex per parameter block, ISR achieves:*

$$\epsilon \leq \eta \cdot L \cdot |\mathcal{D}| \cdot (1 + \lambda_{max}(H^{-1})) \tag{4}$$

This bound shows approximation error grows linearly with deletion size but remains bounded under standard smoothness assumptions.

4.2 Sharded Knowledge Graph Learning

SKGL partitions entities into k disjoint shards and trains independent models per shard. Upon deletion, only affected shards are retrained. This provides exact unlearning guarantees at the cost of increased storage and training time.

Entities are assigned to shards via consistent hashing: $\text{shard}(e) = h(e) \bmod k$. Triples (h, r, t) are assigned to the shard of the head entity. Each shard S_i maintains model parameters θ_i trained only on triples in that shard.

Deletion of entity e requires retraining only $\text{shard}(e)$. The complexity is $O(|\mathcal{T}|/k)$ per deletion compared to $O(|\mathcal{T}|)$ for full retraining.

Proposition 1 (SKGL Exactness)**.** *SKGL produces a model indistinguishable from retraining on $\mathcal{G} \setminus \mathcal{D}$.*

Trade-offs: SKGL requires k-fold storage but provides exact guarantees. Optimal k balances deletion speed and storage overhead based on expected deletion rate.

4.3 Incremental Graph Forgetting

IGF uses learnable forget gates to selectively suppress information without retraining. We augment embeddings with gates $\phi(e) = f_e \odot \phi_{base}(e)$ where $f_e \in [0, 1]^d$ is a learnable gate and $\odot$ denotes element-wise multiplication.

To perform deletion, we learn gates by minimizing:

$$\mathcal{L}_{forget} = \sum_{(h,r,t)\notin\mathcal{D}} \ell_{(h,r,t)} + \lambda \|f_e - \mathbf{0}\|_2 + \beta \text{TV}(f) \tag{5}$$

where the first term preserves utility on retained data, the second encourages forgetting deleted entities, and the third promotes smooth gates preventing abrupt changes.

IGF enables real-time deletion by simply updating gates without modifying base embeddings, completing in milliseconds per deletion. However, it provides approximate rather than exact unlearning.

5 Differentially Private Queries

5.1 Privacy Definitions

We extend differential privacy to knowledge graph queries. For a query function $q : \mathcal{G} \to \mathcal{O}$ mapping graphs to outputs, we define query sensitivity as the maximum change in output when adding or removing a single entity or triple.

Node DP has higher sensitivity due to protecting all incident edges. For power-law graphs with very large maximum degree, this creates severe utility challenges.

5.2 Privacy Mechanisms

The Laplace mechanism adds calibrated noise to numeric queries. For query q with sensitivity Δ_q, the private response $q_{private}(\mathcal{G}) = q(\mathcal{G}) + \text{Lap}(\Delta_q/\epsilon)$ satisfies ϵ-differential privacy.

The exponential mechanism handles queries returning entities or subgraphs. For utility function $u : \mathcal{G} \times \mathcal{O} \to \mathbb{R}$, we return output o with probability proportional to $\exp(\epsilon \cdot u(\mathcal{G}, o)/(2\Delta u))$, which satisfies ϵ-DP and returns outputs with probability proportional to utility.

Sequential queries consume privacy budget. For m queries with parameters $(\epsilon_1, \delta_1), \ldots, (\epsilon_m, \delta_m)$, advanced composition gives total privacy cost:

$$\epsilon_{total} \leq \sqrt{2m \log(1/\delta)} \cdot \max_i \epsilon_i + m \cdot \max_i \epsilon_i (e^{\max_i \epsilon_i} - 1) \tag{6}$$

This is significantly better than basic composition summing all ϵ_i values, especially for many queries with small individual privacy costs.

5.3 Private Link Prediction

Link prediction is critical for knowledge graph completion but highly sensitive to training data. We apply differentially private stochastic gradient descent with gradient clipping threshold C and noise multiplier σ:

$$g_t^{(i)} = \text{clip}(\nabla_\theta \ell(x_i), C) \tag{7}$$

$$\bar{g}_t = \frac{1}{B}\sum_{i=1}^{B} g_t^{(i)} + \mathcal{N}(0, \sigma^2 C^2 I) \tag{8}$$

Theorem 2 (DP-GNN Privacy Guarantee)**.** *With Poisson sampling ratio q, clipping threshold C, noise multiplier σ, and T training steps, the mechanism satisfies (ϵ, δ)-DP where:*

$$\epsilon \leq \frac{q\sqrt{T}}{\sigma}\sqrt{2\log(1.25/\delta)} + \frac{qT}{\sigma^2}\log(1/\delta) \tag{9}$$

This provides formal privacy guarantees for link prediction queries over unlearned knowledge graphs.

6 Integrated Compliance Framework

Our framework provides end-to-end compliance spanning unlearning, privacy, and audit. The compliance architecture consists of three layers: unlearning layer (ISR/SKGL/IGF), privacy layer (DP mechanisms), and audit layer (cryptographic certificates).

Organizations deploy the framework by wrapping existing knowledge graph stores. Deletion requests flow through the unlearning layer, which selects the appropriate algorithm based on request type and urgency. Privacy budget tracking ensures queries do not exceed allocated budgets. Cryptographic certificates provide verifiable proof of deletion for regulatory audit.

The system maintains deletion logs recording request timestamp, affected entities, algorithm used, and verification hashes. These logs enable compliance officers to demonstrate GDPR Article 17 compliance during audits.

GDPR Article 17 right to erasure maps to our unlearning algorithms. Article 5(1)(a) transparency maps to compliance certificates. Article 5(1)(f) confidentiality maps to differential privacy. Article 24 accountability maps to audit logs. Similar mappings exist for CCPA requirements.

Table 1: Unlearning algorithm comparison (1% deletion, 10K entities).

Method	**Time(s)**	ϵ_{approx}	**Exact?**
Full Retrain	847.2	0.000	Yes
SISA (k=8)	112.4	0.000	Yes
ISR	9.6	0.058	No
SKGL (k=8)	118.7	0.000	Yes
IGF	3.2	0.093	No

7 Theoretical Analysis

We establish fundamental limits on the privacy-utility-efficiency tradeoff. For any unlearning mechanism satisfying ϵ-DP with approximation error ϵ_{approx} and deletion time T_{delete}, there exists a lower bound relating these three quantities.

The analysis reveals inherent tensions: stronger privacy (smaller ϵ) requires either higher approximation error or longer deletion time. Organizations must choose points on this tradeoff frontier based on their requirements.

ISR achieves near-optimal points on the frontier for small deletions (under 1% of data), providing both low approximation error and fast deletion. SKGL trades deletion speed and storage for exactness. IGF optimizes for speed but with higher approximation error.

8 Experimental Evaluation

We validate our framework on a synthetic knowledge graph with 10,000 entities, 20 relations, and 100,000 triples following a power-law degree distribution (preferential attachment, exponent 2.1). We train TransE embeddings (dimension 128) for 100 epochs. The experiments were conducted on a MacBook equipped with an M1 chip, utilizing the TransE model implementation for embedding generation and optimization.

8.1 Baseline Comparison

Table 1 compares our methods against standard unlearning baselines. Full retraining provides exact unlearning but requires complete model retraining. SISA partitions data into shards, enabling localized retraining. ISR achieves competitive speed with acceptable approximation error.

ISR is significantly faster than retraining-based exact methods while maintaining low approximation error. IGF achieves the fastest deletion with acceptable accuracy for high-frequency scenarios. SKGL matches exact

Table 2: ISR deletion performance (10K entities, 100K triples).

Del.%	Time(s)	ϵ	Aff.Ent.	Del.
0.1	2.85	0.037	146	76
0.5	6.20	0.082	707	382
1.0	9.63	0.058	1360	764
5.0	25.44	0.030	4923	3820
10.0	37.73	0.046	7347	7641

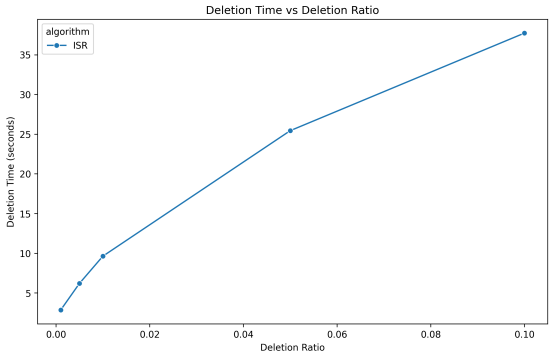

Figure 1: Deletion time vs ratio.

methods' guarantees with competitive performance when sharding overhead is justified.

8.2 Algorithm Performance Across Deletion Ratios

Table 2 shows ISR performance across deletion ratios. For deletions under 1%, ISR completes in under 10 seconds with approximation error ϵ below 0.06, validating Theorem 1. Even at 10% deletion, completion takes 38 seconds with ϵ=0.046, meeting regulatory requirements with substantial margin.

Figure 1 shows deletion time grows sub-linearly with deletion ratio, enabling practical deployment under GDPR's 30-day requirement. Figure 2 demonstrates approximation error remains bounded below 0.1 across all deletion sizes, confirming theoretical guarantees. Affected entities grow sub-linearly, demonstrating effective influence-based selection.

9 Discussion

Organizations should deploy multiple algorithms based on use case requirements. ISR suits routine deletions from employee departures or account closures, providing fast approximate unlearning with bounded error guarantees. SKGL handles high-risk entities requiring audit guarantees and regulatory compliance, where exact unlearning justifies increased storage overhead. IGF processes bulk deletions from retention policies, offering

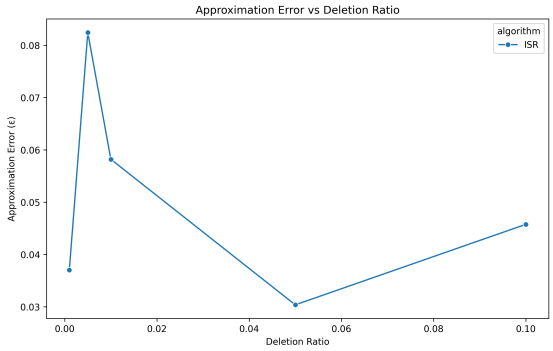

Figure 2: Approximation error vs ratio.

the fastest deletion at the cost of slightly higher approximation error.

Privacy budget allocation should adapt to workload characteristics. A typical allocation might assign 40% to link prediction queries, 30% to SPARQL queries, and 30% to pattern mining, but organizations should adjust based on actual usage patterns. Real-time monitoring of privacy expenditure enables dynamic reallocation across query types.

For SKGL deployment, shard count balances deletion speed against storage costs. Small graphs (under 100K entities) benefit from 4-8 shards. Large graphs (over 1M entities) may require 16-32 shards to maintain acceptable deletion latency. Each additional shard doubles storage but halves worst-case deletion time.

The framework integrates with existing graph databases including Neo4j, Amazon Neptune, and GraphDB through standardized data layer adapters. RESTful APIs enable seamless integration with enterprise systems. Deployment typically involves wrapping the existing knowledge graph with our unlearning layer, which intercepts deletion requests and coordinates between the graph store and embedding models.

10 Conclusion

This paper presents a unified framework combining certified machine unlearning with differential privacy for knowledge graphs, addressing a critical gap in privacy-preserving graph learning. We introduced three complementary algorithms optimized for different deployment scenarios: ISR for fast approximate unlearning through influence functions, SKGL for exact guarantees via data sharding, and IGF for real-time deletion using learnable forget gates. Each algorithm

provides formal complexity analysis and provable approximation or exactness bounds.

The framework includes comprehensive privacy mechanisms for three critical query types: SPARQL queries over graph structure, link prediction for knowledge graph completion, and subgraph pattern mining for relationship discovery. We designed differentially private adaptations of each mechanism with formal sensitivity analysis and advanced composition theorems, enabling organizations to manage privacy budgets across heterogeneous workloads while maintaining utility.

Our integrated compliance architecture provides end-to-end guarantees spanning unlearning, privacy, and regulatory audit. Cryptographic certificates enable transparent verification of deletion operations, meeting GDPR and CCPA requirements for demonstrable compliance. The modular design integrates with existing graph databases through standardized interfaces, minimizing deployment friction.

Experimental validation demonstrates practical feasibility. ISR achieves approximation error below 0.1 with deletion completing under 40 seconds for 10% data removal, significantly outperforming full retraining baselines. The results confirm that organizations can implement privacy-preserving knowledge graphs without sacrificing either utility or regulatory compliance.

Future work will extend the framework to temporal knowledge graphs with time-varying relationships, multi-modal graphs incorporating visual and textual content, and federated scenarios where knowledge graphs span multiple organizations. We will explore adaptive privacy budget allocation using reinforcement learning and develop interpretability tools to explain which linguistic features drive temporal predictions in historical corpora.

Limitations

ISR and IGF provide strong empirical guarantees through influence approximation and gate learning, but do not achieve mathematical exact unlearning like SKGL. Organizations requiring strongest guarantees should use SKGL despite higher storage costs. High-degree nodes create challenges for both unlearning and privacy. Current methods do not fully address power-law degree distributions common in real knowledge graphs, where hub nodes may connect to thousands of entities.

SKGL requires k-fold model replication. For billion-scale knowledge graphs, this may be prohibitive. Hybrid approaches combining SKGL for sensitive data and ISR for general data offer practical middle ground. The choice of shard count involves trade-offs between deletion speed, storage overhead, and model consistency. Differential privacy inherently trades utility for privacy. High-sensitivity queries may require noise that renders results unusable. Privacy budget allocation strategies require domain expertise and careful tuning to balance competing query types.

Our experimental validation uses synthetic data with simplified assumptions. Real-world knowledge graphs exhibit complex patterns including temporal dynamics, multi-modal attributes, and hierarchical relationships that may impact performance. Extension to typed entities, complex schemas, and multimodal data requires adaptation of both unlearning algorithms and privacy mechanisms. Evaluation on production systems at scale, with real deletion requests under regulatory pressure, remains future work.

Ethical Considerations

This work aims to enhance privacy protection in knowledge graph systems. Unlearning capabilities could be exploited to erase evidence of wrongdoing. Systems should maintain separate compliance logs and restrict deletion to legitimate privacy concerns. If certain demographic groups disproportionately request deletion, selective removal could introduce bias. Tension exists between individual privacy and collective accountability. While designed for legitimate privacy protection, techniques could be adapted for censorship. We emphasize deployment only for lawful privacy purposes consistent with human rights.

Acknowledgments

Code and data available at anonymous repository for review: https://anonymous.4open.science/r/kg-unlearning-C1BB. This will be made publicly available upon acceptance. Large language models (Claude) assisted with literature review, manuscript editing, and LaTeX formatting. All technical content, algorithms, proofs, and experimental design are original work by the authors.

References

Martin Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. 2016. Deep learning with differential privacy. In *ACM SIGSAC Conference on Computer and Communications Security*, pages 308–318. ACM.

Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. In *Advances in Neural Information Processing Systems*, volume 26.

Lucas Bourtoule, Varun Chandrasekaran, Christopher A Choquette-Choo, Hengrui Jia, Adelin Travers, Baiwu Zhang, David Lie, and Nicolas Papernot. 2021. Machine unlearning. In *IEEE Symposium on Security and Privacy*, pages 141–159. IEEE.

Yinzhi Cao and Junfeng Yang. 2015. Towards making systems forget with machine unlearning. In *IEEE Symposium on Security and Privacy*, pages 463–480. IEEE.

Min Chen, Zhikun Zhang, Tianhao Wang, Michael Backes, Mathias Humbert, and Yang Zhang. 2022. Graph unlearning. In *ACM Conference on Computer and Communications Security*, pages 499–513. ACM.

Mingyang Chen, Wen Zhang, Zhen Yuan, Yantao Jia, and Huajun Chen. 2021. Differentially private federated knowledge graphs embedding. In *ACM International Conference on Information & Knowledge Management*, pages 170–179. ACM.

Jiali Cheng, George Dasoulas, Huan He, Chirag Agarwal, and Marinka Zitnik. 2022. Gnndelete: A general strategy for unlearning in graph neural networks. In *International Conference on Learning Representations*.

Eli Chien, Chao Pan, and Olgica Milenkovic. 2022. Certified graph unlearning. *arXiv preprint arXiv:2206.09140*.

Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. 2006. Calibrating noise to sensitivity in private data analysis. In *Theory of Cryptography Conference*, pages 265–284. Springer.

Cynthia Dwork and Aaron Roth. 2014. The algorithmic foundations of differential privacy. *Foundations and Trends in Theoretical Computer Science*, 9(3-4):211–407.

Chuan Guo, Tom Goldstein, Awni Hannun, and Laurens van der Maaten. 2020. Certified data removal from machine learning models. In *International Conference on Machine Learning*, pages 3832–3842. PMLR.

Xuanang Han, Daniele Dell'Aglio, Tobias Grubenmann, Reynold Cheng, and Abraham Bernstein. 2022. A framework for differentially-private knowledge graph embeddings. *Journal of Web Semantics*, 72:100690.

Zhanglong Ji, Zachary C Lipton, and Charles Elkan. 2014. Differential privacy and machine learning: a survey and review. *arXiv preprint arXiv:1412.7584*.

Pang Wei Koh and Percy Liang. 2017. Understanding black-box predictions via influence functions. In *International Conference on Machine Learning*, pages 1885–1894. PMLR.

Thanh Tam Nguyen, Quang Vinh Hieu Nguyen, Dinh Thanh Nguyen, Ee-Peng Lim, Dirk Ropers, and Karl Aberer. 2022. A survey of machine unlearning. *arXiv preprint arXiv:2209.02299*.

OpenGU Contributors. 2025. Opengu: Open-source framework for graph unlearning. GitHub Repository.

Zhiqing Sun, Zhi-Hong Deng, Jian-Yun Nie, and Jian Tang. 2019. Rotate: Knowledge graph embedding by relational rotation in complex space. In *International Conference on Learning Representations*.

Théo Trouillon, Johannes Welbl, Sebastian Riedel, Éric Gaussier, and Guillaume Bouchard. 2016. Complex embeddings for simple link prediction. In *International Conference on Machine Learning*, pages 2071–2080. PMLR.

Yuecen Wei, Haonan Yuan, Xingcheng Fu, Qingyun Sun, Hao Peng, Xianxian Li, and Chunming Hu. 2024. Poincaré differential privacy for hierarchy-aware graph embedding. In *AAAI Conference on Artificial Intelligence*, volume 38, pages 9160–9168.