Knowledge Graph Unlearning with Schema

Yang Xiao, Ruimeng Ye, Bo Hui
Department of Computer Science, University of Tulsa
`{yax3417, ruy9945, bohui}@utulsa.edu`

Abstract

Graph unlearning emerges as a crucial step to eliminate the impact of deleted elements from a trained model. However, unlearning on the knowledge graph (KG) has not yet been extensively studied. We remark that KG unlearning is non-trivial because KG is distinctive from general graphs. In this paper, we first propose a new unlearning method based on schema for KG. Specifically, we update the representation of the deleted element's neighborhood with an unlearning object that regulates the affinity between the affected neighborhood and the instances within the same schema. Second, we raise a new task: schema unlearning. Given a schema graph to be deleted, we remove all instances matching the pattern and make the trained model forget the removed instances. Last, we evaluate the proposed unlearning method on various KG embedding models with benchmark datasets. Our codes are available at `https://github.com/NKUShaw/KGUnlearningBySchema`.

1 Introduction

To protect users' concerns about privacy and security, laws such as the European Union's General Data Protection Regulation (GDPR), the California Consumer Privacy Act (CCPA), and Canada's proposed Consumer Privacy Protection Act (CPPA) regulate the usage of personal data in machine learning (ML) and give users the right to withdraw consent to the usage of their data (Biega and Finck, 2021; Regulation, 2016; OAG, 2021). Machine unlearning algorithms (Cao and Yang, 2015; Golatkar et al., 2020; Bourtoule et al., 2019; Marchant et al., 2022; Neel et al., 2021a) aim to proactively eliminate the memory about deleted data from already trained machine learning models.

Graph unlearning (Said et al., 2023; Cheng et al., 2023; Chien et al., 2022) emerges as a crucial method to address data privacy and adversarial attacks on graph data such as social networks. Given the elements such as nodes and edges to be deleted, various approaches (Guo et al., 2020; Wu et al., 2023b,a; Chien et al., 2023) have been proposed to remove the influence of deleted elements on both model weights and neighboring representations.

However, unlearning on knowledge graph (KG) has not yet been extensively studied. We remark that KG unlearning is non-trivial. First, KG has been used to describe open knowledge projects such as Wikidata and YAGO (Wiki, 2024; Pellissier Tanon et al., 2020). These KGs allow both humans and machines to acquire information and derive new knowledge. Factors like scientific opinions (e.g., historical ideas about race), socioculture, or political views can lead to an encoding of social bias. Therefore, it is necessary to provide an interface to remove certain knowledge and eliminate the influence on downstream modules such as reasoning. Second, a knowledge graph is distinctive from general graphs. A KG defines abstract classes and relations of entities in a schema. Lastly, the relation between two entities has semantic meanings where the edge on a general graph is only associated with a weight. Due to the unique structure and information, it is a challenge to generalize a graph unlearning algorithm on KGs.

In this paper, we first propose a KG unlearning method based on schema. Given an entity or a relation to be deleted from the KG, existing graph unlearning methods seek to ensure that the relationship between two entities connected by the deleted component is similar to the relation between two random entities as if the relation does not exist (Cong and Mahdavi, 2022; Ye et al., 2023; Peng et al., 2022). However, we argue that such a strategy is too "aggressive" because the two entities could be indirectly connected through other entities on the KG. Therefore, we propose to define a new target for KG unlearning. Schema, as a high-order meta pattern of KG, contains the type constraint between entities and relations, and it can naturally, be

Proceedings of the 31st International Conference on Computational Linguistics, pages 3541–3546
January 19–24, 2025. ©2025 Association for Computational Linguistics

used to capture the structural and semantic information in context (Ghosh et al., 2020; Ye et al., 2023; Peng et al., 2022; Hui et al., 2022). Intuitively, two instances within a schema are similar to each other. Given a component to be removed, we construct a sub-graph containing affected entities. We extract the schema for the sub-graph and query sub-graphs that have the same schema. Lastly, we update the affected neighborhood's representation based on the queried sub-graphs. Our method is applicable to both entity unlearning and relation unlearning.

Furthermore, we raise a new research problem: *schema* unlearning on KG. Since the schema can constrain the entities and relations on the knowledge base, it is intuitive to remove a set of instances with given constraints on KG upon request. We remark that the schema can be used to extract the instances that concern privacy and stereotypes. For example, Schema (person, is a friend of, person) leads to privacy leakage, and Schema (black American, commits, criminality) is related to racial stereotypes. Existing study shows that social biases are engraved in KG (Kraft and Usbeck, 2022). Given a schema, we propose to extract and remove all instances matching the schema from KG. Similar to removing entities or relations, we update the representations of affected neighborhoods.

2 Proposed Method

Let $G = (E, R, S)$ be a KG, where E and R are the sets of entities and relations in the KG. We use S to denote the set of triples, each of which is (e_h, r, e_t), including the head entity $e_h \in E$, the tail entity $e_t \in E$ and the relation r between e_h and e_t. Given a model $M(G)$ trained on G to associate each entity and relation with a vector in an embedding space H, the user can request to delete a subset of entities E_d or a subset of relations R_d. The straightforward solution is to retrain a new model $M(G/E_d)$ (or $M(G/R_d)$) on the remaining data G/E_d (or G/R_d) from scratch. However, this naive method is time-consuming for frequent deletion requests over large-scale data. Therefore, the goal of an efficient unlearning algorithm is to directly eliminate the effects of deleted data on M.

2.1 Unlearning with Schema

Given an entity $e_d \in E_d$ to be deleted, we first extract k-hop enclosing sub-graph G_u around e_d. Intuitively, if e_d is deleted, the representations of nodes in the k-hop neighborhood need to be updated. For example, in Figure 1, the blue nodes represent the nodes in the 2-hop sub-graph around the deleted entity. For each node on the sub-graph, we use RDF (Resource Description Framework) Schema (*e.g., rdf: Class*) (Wikipedia, 2024) to represent the high-order meta pattern of the node and edge. Similarly, we can extract high-order meta patterns for edges on G_u.Then we can use a schema sub-graph G_s to describe the meta pattern of G_u. For example, *"Da Vinci"* on G_u will be replaced with *"rdf: Person"* on G_s.

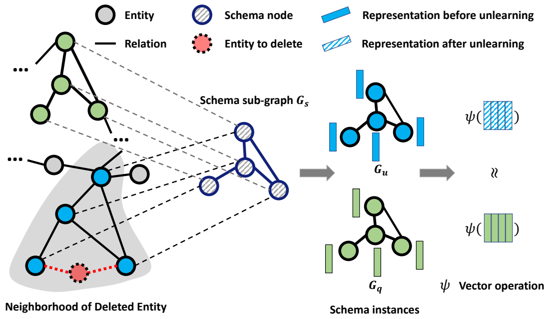


Figure 1: Unlearning with Schema.

With the high-order meta pattern G_s, the next step is to query a sub-graph G_q which also has a high-order meta pattern G_s. Specifically, G_q is isomorphic to G_u and G_q share the schema-graph G_s with G_u. Intuitively, if both G_u and G_q can be described by a schema pattern G_s at high-order, these two sub-graph should be similar to each other. However, sub-graph matching is an NP-complete problem (Lou et al., 2020; Sun et al., 2012). In this paper, we leverage Glasgow Subgraph Solver (McCreesh et al., 2020) to find the sub-graph G_q. To reduce the high computational cost, the solver returns once a sub-graph matches the query instead of finding all sub-graphs. Figure 1 shows an example of the queried G_q (highlighted in green) where all nodes match the pattern on the schema sub-graph. Recall that the target is to update the representation of G_u as if e_d has never existed. For any two entities e_i and e_j on G_u, our target is to maximize the similarity between (e_i, e_j) and (e_m, e_n), where (e_m and e_n) are the corresponding entities on G_q and share the same schema with (e_i, e_j). Specifically, the relation (direct or indirect) in the embedding space between e_i and e_j is supposed to be similar to that between e_i and e_j because they share the same schema sub-graph. Therefore, we maximize the similarity for all pairs on G_u:

$$\sum_{(e_i, e_j \in G_u, e_i \neq e_j)} \frac{(H_i - H_j) \cdot (H_m - H_n)}{||(H_i - H_j)||||(H_m - H_n)||}, \tag{1}$$

where H_i, H_j, H_m, H_n are the embedding of e_i, e_j, e_m, e_n respectively. For any pair (e_i, e_j), we can always find corresponding (e_m, e_n) on G_q. Denote Eq (1) as the unlearning target l_d for the deleted entity e_d. For all deleted entities in E_d, the overall unlearning object is to minimize:

$$\mathcal{L} = \sum_{e_d \in E_d} In(1 - l_d + \epsilon), \qquad (2)$$

where ϵ is a hyperparameter to avoid 0 in $In(\cdot)$.

Similar to deleting an entity, we can construct a neighborhood sub-graph around a deleted $r_d \in R_d$ and leverage Eq (2) to update the representations of the neighborhood around r_d.

2.2 Delete Schema

Note that the schema can constrain the entities and relations on the knowledge base. It provides a way to remove a set of instances with given constraints. For example, we can remove the relations in all instances that match the schema (foaf: Person, rdf: Is a friend of, foaf: Person) to protect privacy. Some data patterns (e.g., a person of type X is a terrorist or a protestor) could have an unwanted impact on downstream modules (e.g., reasoning or classifying if a person is a terrorist), so it is important to remove such patterns in KG. Given a schema pattern to be deleted, there are two solutions to break the pattern: (1) delete a component (e.g., entities or relations) on the instance of a schema sub-graph; (2) remove the whole sub-graph. We remark that the first solution will be transferred to an entity unlearning or relation unlearning problem once the instances are returned. Algorithm 1 describes the second solution to remove the matched sub-graphs.

Algorithm 1 Schema Unlearning
Input: Schema G_s to be deleted, G, H
Output: New G, new embeddings H

```
Query all instances of the query schema $G_s$
Find all sub-graphs $Q = \{G_{q_1}, G_{q_2}, \cdots\}$ matches $G_s$ with Glasgow Solver
for $G_q \in Q$ do
    remove $G_q$ from $G$
    Construct $k$-hop connected sub-graphs around $G_q$.
    for each connected $G_c$ around $G_q$ do
        Maximize Eq (1) for any two entities ($e_i$, $e_i$) on $G_c$
    end for
end for
```

3 Experiments

We evaluated the effectiveness of our unlearning method on three embedding models and compared our method with graph unlearning baselines. We experiment with two datasets: YAGO3-10 (Pellissier Tanon et al., 2020) and FB15k237 (Wang et al., 2019). From each dataset, we sample entities and relations to be deleted, and re-train the embedding model with the remaining data from scratch for comparison. Ideally, the result of unlearning should be similar to re-training on the remaining data. We report Hit@1, Hit@3, Hit@10, and MRR of link prediction task for three embedding models: TransE (Bordes et al., 2013), TransH (Wang et al., 2014), TransD (Ji et al., 2015). Besides the intuitive retraining strategy (), we compare our unlearning method with Gredeint Ascent (Neel et al., 2021b), GIF (Wu et al., 2023a), and the-state-of-the-art baseline GNNDelete (Cheng et al., 2023). Note that GNNDelete outperforms other baselines including GraphEraser (Chen et al., 2022) and GraphEditor (Cong and Mahdavi, 2022) in terms of both accuracy and efficiency (Cheng et al., 2023). In the experiment, We follow (Ye et al., 2023) to randomly choose schemas to be deleted. For entity unlearning and relation unlearning, we randomly delete components (i.e., entities, relations) to observe the results after unlearning.

Results and analysis Table 1 shows the performance of the link prediction before deleting entities (labeled as "original") and after unlearning. Ideally, the result after unlearning should be close to "retraining". We have removed about 10% entities randomly from the dataset. Compared with other unlearning baselines, we can observe that the performance of our unlearning is closer to "retraining" in terms of all performance metrics. Interestingly, none of these baseline methods have comparable performance to our method on these performance metrics. These baseline unlearning methods lead to drastic performance degradation and lose almost the prowess in making meaningful predictions. It further verifies that existing unlearning methods are too "aggressive". Compared with general graphs, the knowledge graph is more complicated because there are semantic relations between entities. Only considering the direct connection between entities on the graph may ignore intrinsic connection after deleting components. We also examine deleting relations and schemas in Table 2 and 3. The conclusion still holds for deleting relations and schemas.

Model	Method	YAGO				FB15k			
		Hit@10	Hit@3	Hit@1	MRR	Hit@10	Hit@3	Hit@1	MRR
TransE	Original	0.5443	0.3887	0.2189	0.3315	0.4764	0.3253	0.1939	0.2892
	Retrain	0.5080	0.3661	0.2058	0.3111	0.4416	0.3016	0.1774	0.2672
	Gradient Ascent	0.3623	0.1248	0.0515	0.1353	0.3394	0.2198	0.1236	0.1814
	GNNDelete	0.3713	0.2321	0.1163	0.2012	**0.4147**	0.2785	0.1669	0.2498
	GIF	0.4479	0.2649	0.0896	0.2109	0.3394	0.1941	0.1034	0.1806
	Our Method	**0.5107**	**0.3443**	**0.1814**	**0.2933**	0.4744	**0.3230**	**0.1854**	**0.2831**
TransH	Train	0.6148	0.4773	0.3015	0.4124	0.4844	0.3337	0.2018	0.2967
	Retrain	0.5615	0.4305	0.2716	0.3725	0.4497	0.3026	0.1704	0.2647
	Gradient Ascent	0.3706	0.1174	0.0509	0.1327	0.3416	0.1949	0.1073	0.1833
	GNNDelete	0.3459	0.2223	0.1155	0.1944	0.3444	0.2262	0.1338	0.2045
	GIF	0.4283	0.2467	0.0335	0.1706	0.4080	0.2652	0.1422	0.2319
	Our Method	**0.5519**	**0.4003**	**0.2274**	**0.3384**	**0.4527**	**0.2901**	**0.1554**	**0.2529**
TransD	Train	0.6011	0.4543	0.2791	0.3915	0.4840	0.3302	0.1976	0.2931
	Retrain	0.5512	0.4168	0.2617	0.3626	0.4502	0.3016	0.1643	0.2614
	Gradient Ascent	0.3690	0.1192	0.0509	0.1331	0.3403	0.1916	0.1050	0.1813
	GNNDelete	0.3613	0.2286	0.1116	0.1971	0.3108	0.1946	0.1123	0.1783
	GIF	0.4753	0.2973	0.0451	0.1997	0.3665	0.2279	0.1139	0.1984
	Our Method	**0.5751**	**0.4197**	**0.2390**	**0.3546**	**0.4572**	**0.2850**	**0.1399**	**0.2443**

Table 1: Delete entities (about 10% entities) on YAGO and FB15K-237

Model	Method	Hit@10	Hit@3	Hit@1	MRR
TransE	Original	0.5443	0.3887	0.2189	0.3315
	Retrain	0.5232	0.3770	0.2199	0.3245
	Gradient Ascent	0.3541	0.1199	0.0408	0.1262
	GNNDelete	0.4413	0.2904	0.1604	0.2543
	GIF	0.4613	0.2806	0.0840	0.2134
	Our Method	**0.5256**	**0.3751**	**0.2126**	**0.3216**
TransH	Train	0.6148	0.4773	0.3015	0.4124
	Retrain	0.5901	0.4600	0.3008	0.4018
	Gradient Ascent	0.3600	0.1054	0.0362	0.1193
	GNNDelete	0.4427	0.2966	0.1639	0.2580
	GIF	0.4370	0.2605	0.0343	0.1774
	Our Method	**0.5965**	**0.4655**	**0.2991**	**0.4042**
TransD	Train	0.6011	0.4543	0.2791	0.3915
	Retrain	0.5764	0.4408	0.2773	0.3820
	Gradient Ascent	0.3587	0.1075	0.0363	0.1198
	GNNDelete	0.4694	0.3157	0.1739	0.2732
	GIF	0.4961	0.3171	0.0436	0.2010
	Our Method	**0.5865**	**0.4454**	**0.2788**	**0.3861**

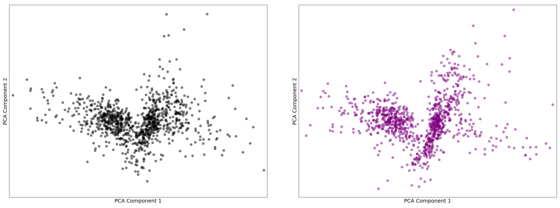
Table 2: Delete relations (about 7% triplets) on YAGO

Time and Space Efficiency. Our unlearning method is both time-efficient and space-efficient as compared to the unlearning baselines. For example, our unlearning method takes about 24 minutes to unlearn 10% entities on TransE while GNNDelete takes about 1 hour and 11 minutes. The GPU memory required for GIF is 50 G and the GPU memory occupied by our method is less than 2 G.

Visualization We project the embeddings of random entities from the dataset "YAGO" in 2-dimensional space for visualization. Figure 2 shows 200 random entities before unlearning and after unlearning. We can see that some embedding will change significantly after unlearning while the overall distribution does not change.

Model	Method	Hit@10	Hit@3	Hit@1	MRR
TransE	Original	0.5443	0.3887	0.2189	0.3315
	Retrain	0.4983	0.3429	0.1724	0.2850
	Gradient Ascent	0.3628	0.1204	0.0429	0.1285
	GNNDelete	0.3725	0.2300	0.105	0.1969
	GIF	0.4432	0.2546	0.0874	0.2055
	Our Method	**0.5038**	**0.3407**	**0.1779**	**0.2887**
TransH	Train	0.6148	0.4773	0.3015	0.4124
	Retrain	0.5223	0.3821	0.2055	0.3190
	Gradient Ascent	0.3730	0.1069	0.0385	0.1233
	GNNDelete	0.3528	0.2213	0.1062	0.1917
	GIF	0.4213	0.2353	0.0354	0.1671
	Our Method	**0.5407**	**0.3890**	**0.2207**	**0.3304**
TransD	Train	0.6011	0.4543	0.2791	0.3915
	Retrain	0.5214	0.3779	0.1988	0.3128
	Gradient Ascent	0.3707	0.1081	0.0379	0.1229
	GNNDelete	0.3725	0.2300	0.1051	0.1969
	GIF	0.4649	0.2764	0.0446	0.1910
	Our Method	**0.5719**	**0.4101**	**0.2353**	**0.3505**

Table 3: Delete schemas (about 10% triplets) on YAGO



(a) Original (b) Unlearning

Figure 2: Visualization of unlearning

4 Conclusion

In this paper, we propose a new unlearning method based on schema for knowledge graph. Given components to be deleted, we update the neighborhood representation with sub-graphs within the same schema. The experiment verifies that our method outperforms the baselines.

Limitations

This paper focuses on an important task: deleting components from KGs and eliminating their influence on downstream modules. We do not make any statements regarding its performance beyond this scope. One limitation of our work is that we only measure the performance regarding link prediction. The deletion requests are supposed to be approved before deletion.

References

Asia J Biega and Michèle Finck. 2021. Reviving purpose limitation and data minimisation in data-driven systems. *arXiv preprint arXiv:2101.06203*.

Antoine Bordes, Nicolas Usunier, Alberto García-Durán, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. In *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States*, pages 2787–2795.

Lucas Bourtoule, Varun Chandrasekaran, Christopher A. Choquette-Choo, Hengrui Jia, Adelin Travers, Baiwu Zhang, David Lie, and Nicolas Papernot. 2019. Machine unlearning. *CoRR*, abs/1912.03817.

Yinzhi Cao and Junfeng Yang. 2015. Towards making systems forget with machine unlearning. In *2015 IEEE Symposium on Security and Privacy, SP 2015, San Jose, CA, USA, May 17-21, 2015*, pages 463–480. IEEE Computer Society.

Min Chen, Zhikun Zhang, Tianhao Wang, Michael Backes, Mathias Humbert, and Yang Zhang. 2022. Graph unlearning. In *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security, CCS 2022, Los Angeles, CA, USA, November 7-11, 2022*, pages 499–513. ACM.

Jiali Cheng, George Dasoulas, Huan He, Chirag Agarwal, and Marinka Zitnik. 2023. Gnndelete: A general strategy for unlearning in graph neural networks. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

Eli Chien, Chao Pan, and Olgica Milenkovic. 2022. Certified graph unlearning. *CoRR*, abs/2206.09140.

Eli Chien, Chao Pan, and Olgica Milenkovic. 2023. Efficient model updates for approximate unlearning of graph-structured data. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

Weilin Cong and Mehrdad Mahdavi. 2022. Grapheditor: An efficient graph representation learning and unlearning approach.

Subhasis Ghosh, Arpita Kundu, Aniket Pramanick, and Indrajit Bhattacharya. 2020. Discovering knowledge graph schema from short natural language text via dialog. In *Proceedings of the 21th Annual Meeting of the Special Interest Group on Discourse and Dialogue, SIGdial 2020, 1st virtual meeting, July 1-3, 2020*, pages 136–146. Association for Computational Linguistics.

Aditya Golatkar, Alessandro Achille, and Stefano Soatto. 2020. Eternal sunshine of the spotless net: Selective forgetting in deep networks. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pages 9301–9309. Computer Vision Foundation / IEEE.

Chuan Guo, Tom Goldstein, Awni Y. Hannun, and Laurens van der Maaten. 2020. Certified data removal from machine learning models. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 3832–3842. PMLR.

Bo Hui, Tian Xia, and Wei-Shinn Ku. 2022. A localized geometric method to match knowledge in low-dimensional hyperbolic space. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*, pages 2822–2832. Association for Computational Linguistics.

Guoliang Ji, Shizhu He, Liheng Xu, Kang Liu, and Jun Zhao. 2015. Knowledge graph embedding via dynamic mapping matrix. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, ACL 2015, July 26-31, 2015, Beijing, China, Volume 1: Long Papers*, pages 687–696. The Association for Computer Linguistics.

Angelie Kraft and Ricardo Usbeck. 2022. The lifecycle of "facts": A survey of social bias in knowledge graphs. In *Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing, AACL/IJCNLP 2022 - Volume 1: Long Papers, Online Only, November 20-23, 2022*, pages 639–652. Association for Computational Linguistics.

Zhaoyu Lou, Jiaxuan You, Chengtao Wen, Arquimedes Canedo, Jure Leskovec, et al. 2020. Neural subgraph matching. *arXiv preprint arXiv:2007.03092*.

Neil G Marchant, Benjamin IP Rubinstein, and Scott Alfeld. 2022. Hard to forget: Poisoning attacks on certified machine unlearning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 7691–7700.

Ciaran McCreesh, Patrick Prosser, and James Trimble. 2020. The glasgow subgraph solver: Using constraint programming to tackle hard subgraph isomorphism problem variants. In *Graph Transformation - 13th International Conference, ICGT 2020, Held as Part of STAF 2020, Bergen, Norway, June 25-26, 2020, Proceedings*, volume 12150 of *Lecture Notes in Computer Science*, pages 316–324. Springer.

Seth Neel, Aaron Roth, and Saeed Sharifi-Malvajerdi. 2021a. Descent-to-delete: Gradient-based methods for machine unlearning. In *Algorithmic Learning Theory*, pages 931–962. PMLR.

Seth Neel, Aaron Roth, and Saeed Sharifi-Malvajerdi. 2021b. Descent-to-delete: Gradient-based methods for machine unlearning. In *Algorithmic Learning Theory, 16-19 March 2021, Virtual Conference, Worldwide*, volume 132 of *Proceedings of Machine Learning Research*, pages 931–962. PMLR.

CA OAG. 2021. Ccpa regulations: Final regulation text. *Office of the Attorney General, California Department of Justice*, page 1.

Thomas Pellissier Tanon, Gerhard Weikum, and Fabian Suchanek. 2020. Yago 4: A reason-able knowledge base. In *The Semantic Web: 17th International Conference, ESWC 2020, Heraklion, Crete, Greece, May 31–June 4, 2020, Proceedings 17*, pages 583–596. Springer.

Miao Peng, Ben Liu, Qianqian Xie, Wenjie Xu, Hua Wang, and Min Peng. 2022. Smile: Schema-augmented multi-level contrastive learning for knowledge graph link prediction. In *Findings of the Association for Computational Linguistics: EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*, pages 4165–4177. Association for Computational Linguistics.

Protection Regulation. 2016. Regulation (eu) 2016/679 of the european parliament and of the council. *Regulation (eu)*, 679:2016.

Anwar Said, Tyler Derr, Mudassir Shabbir, Waseem Abbas, and Xenofon D. Koutsoukos. 2023. A survey of graph unlearning. *CoRR*, abs/2310.02164.

Zhao Sun, Hongzhi Wang, Haixun Wang, Bin Shao, and Jianzhong Li. 2012. Efficient subgraph matching on billion node graphs. *arXiv preprint arXiv:1205.6691*.

Minjie Wang, Da Zheng, Zihao Ye, Quan Gan, Mufei Li, Xiang Song, Jinjing Zhou, Chao Ma, Lingfan Yu, Yu Gai, Tianjun Xiao, Tong He, George Karypis, Jinyang Li, and Zheng Zhang. 2019. Deep graph library: A graph-centric, highly-performant package for graph neural networks. *arXiv preprint arXiv:1909.01315*.

Zhen Wang, Jianwen Zhang, Jianlin Feng, and Zheng Chen. 2014. Knowledge graph embedding by translating on hyperplanes. In *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence, July 27 -31, 2014, Québec City, Québec, Canada*, pages 1112–1119. AAAI Press.

Wiki. 2024. Knowledge graph — Wikipedia, the free encyclopedia. [Online; accessed 2-June-2024].

Wikipedia. 2024. Rdf schema — Wikipedia, the free encyclopedia. [Online; accessed 13-June-2024].

Jiancan Wu, Yi Yang, Yuchun Qian, Yongduo Sui, Xiang Wang, and Xiangnan He. 2023a. GIF: A general graph unlearning strategy via influence function. In *Proceedings of the ACM Web Conference 2023, WWW 2023, Austin, TX, USA, 30 April 2023 - 4 May 2023*, pages 651–661. ACM.

Kun Wu, Jie Shen, Yue Ning, Ting Wang, and Wendy Hui Wang. 2023b. Certified edge unlearning for graph neural networks. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, KDD 2023, Long Beach, CA, USA, August 6-10, 2023*, pages 2606–2617. ACM.

Hongbin Ye, Honghao Gui, Xin Xu, Xi Chen, Huajun Chen, and Ningyu Zhang. 2023. Schema-adaptable knowledge graph construction. In *Findings of the Association for Computational Linguistics: EMNLP 2023, Singapore, December 6-10, 2023*, pages 6408–6431. Association for Computational Linguistics.