Supply Chain Link Prediction on Uncertain Knowledge Graph

Nils Brockmann[1]*, Edward Elson Kosasih[1] *, Simon Baker[2], Iain Blair[2], and Alexandra Brintrup[1]

[1] Institute for Manufacturing, University of Cambridge
[2] Versed AI Ltd.

Abstract. With manufacturing companies outsourcing to each other, multi-echelon supply chain networks emerge in which risks can propagate over multiple entities. Considerable structural and organizational barriers hamper obtaining the supply chain visibility that would be required for a company to monitor and mitigate these risks. Our work proposes to combine the automated extraction of supply chain relations from web data using NLP with augmenting the results using link prediction. For this, the first graph neural network based approach to model uncertainty in supply chain knowledge graph reasoning is shown. We illustrate our approach on a novel dataset and manage to improve the state-of-the-art performance by 60% in uncertainty link prediction. Generated confidence scores support real-world decision-making. This is a work-in-progress paper.

1 Introduction

Outsourcing value generation can bring a competitive advantage to a manufacturing company. The impact of this practice has even greater potential once supplier relationships emerge beyond national borders. The benefits of a global supply chain include access to more favourable buying conditions for raw materials, cheaper labour, and arbitrage opportunities [1]. With supplier companies engaging in outsourcing themselves, multi-echelon and globally distributed supply chains emerge.

The interdependency that results from obtaining goods and services from another firm exposes a company to a multitude of supply chain risks, as illustrated in Figure 1. A low-frequency-high-impact event can cause a disruption to a higher-tier supplier and, as a result of manufacturers depending on upstream entities, propagate through a supply chain [2]. This is known as the ripple effect [3]. Similarly, the chain liability effect occurs when a company suffers damage to its reputation through an upstream entity as the result of consumers not differentiating between individual members of a supply chain in matters of ESG misconduct [4]. It is further possible that a complex supply chain structure obscures the concentration in the production of a critical sub-component or material.

Supply Chain Risk Management (SCRM) is entrusted with reducing a firm's vulnerability to supply chain risks. Decisive for the success of SCRM is the supply chain visibility that a company can obtain. Whereas a manufacturer naturally has a certain oversight over the operations of its direct suppliers, this might not necessarily be true beyond that. Increasing supply chain visibility requires the collaboration of partners. In

* equal contribution

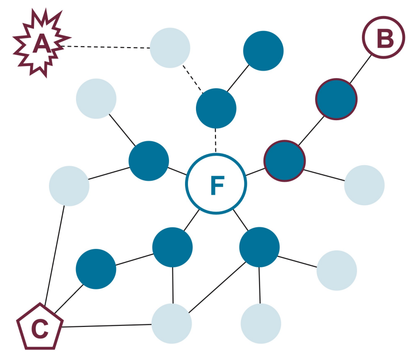

Fig. 1: Supply Chain Network around a focal firm F with visible suppliers in dark blue and three type of risks; A: Ripple Effect, B: Chain Liability Effect, C: Critical Concentration

practice, organizational and structural challenges often prohibit that [5]. Approaches such as track-and-trace RFID and blockchain thus have only limited viability. To address this, a company can engage in Digital Supply Chain Surveillance (DSCS) and obtain knowledge about its supply chain network by proactively monitoring digital data [6].

In a recent DSCS approach, researchers have shown that multiple types of supply chain relations can be extracted from unstructured web data using Natural Language Processing (NLP) [7]. The result is a graph-structure mapping of a supply chain network with confidence scores expressing the classification uncertainty of the Machine Learning model attached to its different relations. This form of data representation is known as an uncertain knowledge graph. Another approach to DSCS uses link prediction on an existing but incomplete graph representation of a supply chain network. This was first realised by [8] considering relational features of entities that were identified as relevant based on domain knowledge. Limitations of this approach due to the required knowledge were addressed by [9] using Graph Neural Networks to automatically encode a node's neighbourhood. This work was extended to knowledge graphs by [10] and [11].

The link prediction approaches as described above could bring a considerable benefit when applied to the supply chain data that was cost-effectively extracted using NLP. This, however, is not possible as they lack the capability to encode the inherent uncertainty of the extracted relation. Whereas there is literature on embedding uncertain knowledge graphs in the wider computer science domain [12], there has not been an application in a supply chain context. This paper proposes to fill this gap and use GNN to perform link prediction on an uncertain supply chain knowledge graph. In particular, our two contributions are:

1. Combining NLP and knowledge graph reasoning.
2. Performing link prediction on a uncertain graph in a supply chain context

The paper is structured as follows. Chapter 2 discusses existing literature on enhancing Supply Chain Visibility using ML as well as giving an introduction to knowledge graphs and their embedding with attached uncertainty. Chapter 3 introduces our proposed approach. Chapter 4 presents the dataset from VersedAI and the results of our experiments. Lastly, Chapter 5 proposes limitations and future research directions.

2 Background and Related Works

2.1 ML to enhance supply chain visibility

Supply Chain Visibility can be enhanced by creating a systematic mapping of company relations. [7] proposed a way in which this can be done by inferring supply chain relations from publicly available web data using NLP. To do this, a classification task was performed on text corpus taken from websites to decide whether they imply a relationship between two companies. Furthermore, different company-to-company affiliations were identified like supplier relationships or the ownership of an entity by another. Before supply chain relations could be inferred, manual training of the classifier was required. Relationship data generated in this way has uncertainty about a classification being correct attached to it which can be expressed by a confidence score.

Supply chain visibility can also be enhanced by predicting unknown relationships between companies based on existing but incomplete knowledge of the supply chain network. This was first done by [8] with the Supply Network Link Predictor (SNLP) model. For this approach, a classifier was trained on four relational features that were deemed relevant for the likelihood of an existing supplier relationship by domain experts. Although this method showed promising results, it relies on the manual selection of features. Obtaining such specialised knowledge can involve considerable costs and the insights are not necessarily transferable to other fields. To alleviate the need for manually selected features, [9] introduced Graph Neural Networks for an automated approach to link prediction. This special form of Neural Network works on graph data and can consider a node's neighbourhood when creating the latent vector representation of it. This allows the model to learn from a company's position in the supply chain network without the need for manually defined relational features. [10] developed this approach further by using GNNs on knowledge graphs containing supply network information. A knowledge graph features multiple types of edges to encode information beyond simple buyer-seller relationships like the affiliation of a company with producing a certain item. As a result, more complex queries can be facilitated based on this graph structure.

2.2 Knowledge Graphs

A knowledge graph G is a multi-relational and directed graph structure [13](Figure 2A). It is made of a set of facts F represented by triplets (h, r, t) that consist of a head entity h, a tail entity t and a connecting relation r. The graph can be denoted as $G = (E, R, F)$ with E and R being the sets of included entities and relations. Knowledge graphs are a structured representation of knowledge. Depending on their type, entities can be real-world objects or concepts. In a supply chain context, this means facts can describe, for example, a supplier relationship between two real-world objects like companies or an affiliation of a company with a concept like a business sector. In an uncertain knowledge graph the triplets (h, r, t, s) are weighted by a confidence score s.

2.3 Uncertain Knowledge Graph Embeddings

The objective of Graph Representation Learning is to encode the nodes and, in some cases, also the edges of a graph in low-dimensional space. This should be done whilst preserving structural information and, in the case of uncertain knowledge graphs,

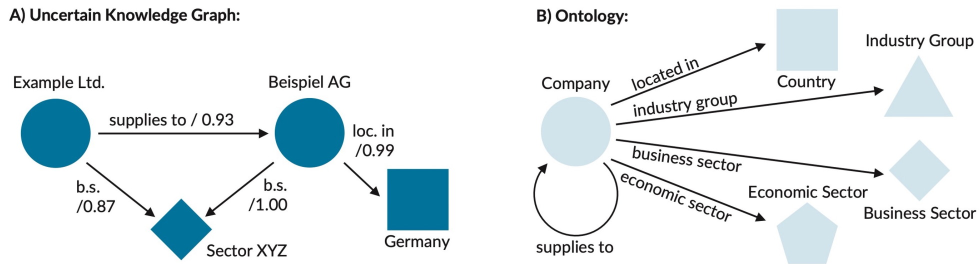

Fig. 2: Uncertain Supply Chain Knowledge Graph with corresponding Ontology

also confidence scores. For deterministic knowledge graphs, there is an abundance of embedding models including TransE [14], with the group of models derived from it referred to as TransX, and ComplEx [15]. Similar, uncertain graphs can be embedded using models like URGE [16]. The selection of models for embedding uncertain knowledge graphs, however, is still limited with the first algorithm of this kind, UKGE, proposed by [12].

UKGE embeds all three elements of a triplet as vectors and uses Probabilistic Soft Logic to infer training label scores for sampled negative edges [12]. To derive a plausibility score, the product of the embedding vectors is calculated. PrTransX also embeds all triplet elements but uses the geometric distance between them to initially calculate a not normalised plausibility score [17]. An alternative approach is BUErRE, a model that embeds only the head and tail entities as hyper-rectangles allowing it to use their intersection as a proxy for the plausibility score [18]. In all three models, the plausibility scores are normalised to a confidence score before the loss is calculated. The uncertain knowledge graph embedding model that shows the best results when tested on benchmark sets is UKGEbert [19]. Similar to its predecessor UKGsE [20] a transformer is used for the embedding of the triplets that are interpreted as three-word sentences. The corresponding confidence score is learned in an LSTM. An exception among the uncertain knowledge graph embedding models is FocusE [21], an add-on layer that allows deterministic knowledge graph embedding models to include uncertainty by using it as a weight when calculating loss.

While uncertain knowledge graph embeddings have been explored in other domains, these methods have not been applied to a supply chain knowledge graph.

2.4 Research Gaps

Our literature review has shown that while there have been several supply chain link prediction works, some gaps are remaining, especially when comparing research in supply chain knowledge graph embedding compared to the wider computer science domain. Our paper addresses the gap in uncertainty modelling in knowledge graphs. In particular, our two contributions are as follows.

1. Link prediction performed on an uncertain knowledge graph (obtained from NLP data extraction) representing a supply chain network
2. Embedding an uncertain knowledge graph considering node neighbourhood and topology, to perform inductive link prediction

3 Our Approach

3.1 Benchmark Model

This approach is based on the graph neural network model that was proposed in [10] and [11]. The model not only shows good performance on supply chain knowledge graphs but also has the ability to encode a nodes surrounding topology. For this reason, this model was chosen over established uncertain knowledge graph embedding models that do not possess this characteristic. This model is made of an encoder and decoder. The former is a heterogeneous, multi-relational GNN where weights are learned for every relation type, with aggregation function similar to [22] and [23]. While the latter is a doot product of the head and tail embeddings output fed into a sigmoid function to minimise Binary Cross Entropy loss.

3.2 Modification of the graph convolution model

To address the research gaps listed above the model must be modified to facilitate the embedding of an uncertain knowledge graph. Further changes were made to improve the performance of the model and to estimate how relevant the contribution is to operational SCRM.

Regression A Classification task is appropriate for link prediction on a deterministic graph as the structure indicates the existence of relations between entities. The aim of the model, therefore, is to decide whether there is a relation or not. In an uncertain knowledge graph, however, information about the likelihood of a relation is encoded. Predicting priorly unknown edges must therefore be done by estimating the likelihood of their existence. This makes the link prediction a regression task. Similar to other models the likelihood is quantified by a probability and thus the values range between 0 and 1. The change from a classification to a regression task requires an adjustment to how loss is calculated. The model described above uses Binary Cross-Entropy Loss and whereas this is common practice for classification tasks the regression requires a different loss function. Staying in line with the objective of the regression, the Mean Squared Error (MSE) is chosen to determine the loss.

ReLU and Cosine Similarity The combination of decoding with the dot product and normalising using a sigmoid function was found to be problematic for the regression. This is due to the strong bias of a sigmoid function for giving values close to its horizontal asymptotes, in this case, 0 and 1. As an alternative, the cosine-similarity was used as a decoder. The cosine similarity between two vectors always belongs to the interval [-1,1]. To obtain values between 0 and 1, as desired to represent a probability, the ReLU-function was used after each convolutional layer on all elements of an embedding vector. This forces all vectors to be located in the first quadrant. The maximum angle between vectors with this constraint is $\frac{\pi}{2}$ restricting the output values to the interval [0,1]. Regardless of the entity types in the triplet, only a single type of convolutional layer was used. We chose a layer that follows the algorithm as proposed by [24].

Entity Type	Number	Edge Type	Number
Company	13770	supplies to	69779
Country	180	domiciled in	7235
Industry Group	61	industry group	5364
Business Sector	32	business sector	5364
Economic Sector	13	economic sector	5364

Table 1: Number of Unique Graph Features

Ensembling Among the multiple approaches that were trialled to increase the performance of the model, ensembling showed the best results. This is also aligned with existing works in the knowledge graph literature such as molecule property prediction [25]. Ensembling is the combination of weak learners to increase performance. In this case, two parallel GNNs of an identical architecture were implemented to generate embeddings independent from each other. The various ways how the embeddings can be combined are discussed with the results in the next chapter.

4 Experiments and Results

4.1 Dataset

For the experiments, an anonymised real-world supply chain dataset extracted using NLP was provided by Versed AI Limited, a supply chain analytics firm based in the United Kingdom. 43.000 facts in the dataset encode supply chain information for more than 13.000 companies. In the dataset, a distinction is made between five entity types (Company, Country, Industry Group, Business Sector, Economic Sector) and five relation types (supplies to, domiciled in, industry group, business sector, economic sector). Following the ontology illustrated in Figure 2B a knowledge graph can be constructed. Table 1 gives the number of unique graph features for each entity and relation type. An important characteristic of the dataset is that it is not limited to a specific industry or geographic region.

As a result of a classifier being used on natural language text to extract the given triplets from web data, there is uncertainty attached to some of them. This is the case for the majority of triplets containing information about a supplier relationship between two companies. All other triplets do not have an uncertainty value attached to them and were assumed to be "certain". These triplets were given an uncertainty value of 1 causing a peak in the histogram illustrating the distribution of the values as shown in Figure 3.

4.2 Experiment Preparations

In order to prepare the data for the experiments, first, a fraction of the nodes (20%) and all edges connecting to them are excluded to be used later in testing the inductive link prediction performance. The remaining triplets are split into three sets, an initial graph (60%), the training set (30%) and the testing set (10%). The set of triplets forming the initial graph is required by the model to determine which nodes are connected and therefore how embeddings are convoluted. The training set is used to train the

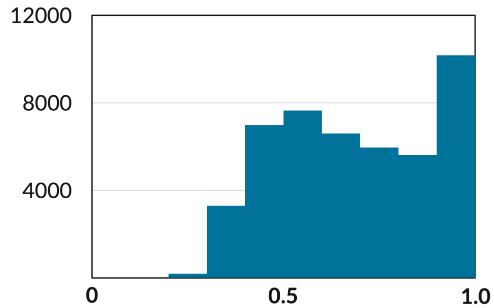

Fig. 3: Histogram of uncertainty values attached to triplets of the type (Company, supplies to, Company)

learnable weights of the model and the test set to measure transductive link prediction performance. Transductive performance is measured by predicting confidence values for unseen edges between nodes that are part of the initial graph. In contrast, inductive performance is the prediction accuracy on unseen edges between those nodes that were initially excluded from the graph. To train the model to differentiate between existing and non-existing edges, negative triplets are sampled. Negative triplets are created by corrupting the tail of a positive triplet and given an uncertainty value of 0. The new tail entity is selected among an entity type which is already present in another triplet containing this head entity to retain the underlying ontology. Within this constraint, the maximum number of negative triplets connecting from one node can be varied. For the experiments, a maximum positive to negative ratio of 1:4 was chosen.

Mean Squared Error (MSE) and Mean Absolute Error (MAE), the commonly used standard metrics for regression, were calculated to quantify the model performance. In the following, MSE is chosen for the comparison of the models instead of MAE as it has a higher tolerance for small divergences and reacts more sensitive to outliers. This is assumed to be preferable from a real-world perspective because small divergences in the predictions would not harm the quality of decision-making to an extent in which an outlier in the prediction could. Similarly, the comparison is made based on the inductive link prediction performance. This is done based on the assumption that link prediction applied in SCRM would have to deal with a changing supply chain network in which companies join and leave.

All experiments were repeated ten times with randomly determined splits of edges and triplets.

4.3 Results

In this section, we compare the performance of the benchmark approach on the VersedAI dataset, with our proposed models. The results are shown in Figure 4.

Benchmark Model As expected, using the benchmark model to inductively predict uncertainty scores for unseen triplets leads to weak results. A MSE of 0.228 indicates no significant difference from what could be expected from a random guess. An explanation for this decrease in performance compared to the good results obtained in a classification

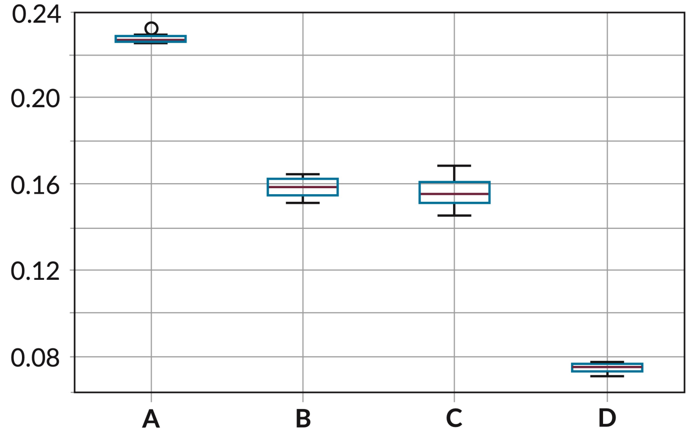

Fig. 4: Mean Squared Error for A: Benchmark; B: Cosine-Similarity with ReLU; C: Standard Ensembling; D: AB-Ensembling

task is the normalisation of scores using a sigmoid function. Whereas the bias of the sigmoid function towards its horizontal asymptotes works in favour of a classification with only zeros and ones as targets, it makes it hard to output values between that as required by the regression.

Cosine Similarity with ReLU Addressing the unfavorable bias introduced by the sigmoid function, cosine similarity was used on embedding vectors that were restricted to be located in the first quadrant by an elementwise application of the ReLU function. MSE decreases by 30% to 0.159 in this approach. Also contributing to the improvement could be cosine-similarity being used as decoder. With the dot-product calculated as $||a||$ $||b||$ $\cos(\Theta)$ it is possible that this leads to a conflict whether to optimise the embedding vectors magnitude or the angle between them.

Standard Ensembling There is empirical evidence that ensemble learning can increase predictive performance but to the best of the authors' knowledge, this method has not yet been used in a supply chain context. In this approach, the plausibility score of two identical GNNs was averaged. However, applied to the model using cosine similarity and ReLU it only leads to a non-significant decrease of the MSE to 0.156 and an increase in the standard deviation from 46 to 68 basis points.

AB Ensembling In another approach to ensemble learning, the model uses the output of the two identical GNNs but combines them before a final score is calculated. Both GNNs generate an embedding for the tail and the head entity of a triplet. When combining the head entity from GNN A with the tail entity from GNN B and vice versa to calculate the confidence score, the MSE drops to 0.075 with a standard deviation of 18 basis points.

4.4 Managerial Insights

We have shown that using our model, we obtain a more accurate link prediction score. Unlike existing literature, our score spans between 0 and 1 (not binary), hence supply chain risk managers can obtain more information about the certainty of the model output. For instance, if the output is close to 0.5 (uncertain), the risk managers can decide to not fully trust the model. This is a desirable feature of a supply chain decision support system.

5 Conclusions and Future Work

Outsourcing value generation leaves a company vulnerable to risks that primarily affect its direct and indirect suppliers. Monitoring these risks becomes increasingly difficult the more complex and geographically distributed a supply chain network is. This is partly because obtaining insight into the wider supply chain network usually requires the collaboration of other firms in the network. An incentive for this is not always present. Digital Supply Chain Surveillance (DSCS) tries to address this issue by using data to gain insights about the network without relying on third-party collaboration.

One approach to DSCS is the extraction of supply chain network information from publicly available web data. This is facilitated by using ML to automatically infer relations between entities from text snippets. Because these entities can not only be companies but also concepts like economic sectors, a knowledge graph can be created based on the extracted data. Another approach is, again using ML, to predict links based on existing but incomplete knowledge of the supply chain network. Previous work shows that link prediction can also be applied successfully on supply chain knowledge graphs.

Both approaches described above could be combined to maximise their impact. However, the data extracted using NLP has uncertainty attached to it resulting in an uncertain knowledge graph created from it. To the best of the authors' knowledge, there has been no work done performing a link prediction on an uncertain supply chain knowledge graph. This work contributes by proposing an ML model that can successfully augment uncertain knowledge graphs representing supply chain network knowledge. Using a GNN architecture for the model also shows how the surrounding topology of a node can be considered when embedding an uncertain knowledge graph.

Experiments were done using a real-world dataset that was automatically extracted from web data using NLP. The Experiments show that missing links in the supply chain knowledge graph can be predicted with an MSE of only 0.075 even if the connected entities were not known at the time of training the model. This has the potential to significantly support decision-making in SCRM.

We propose two directions for future works. First, transformed-based models could be explored that have shown to outperform other models on benchmark datasets. However, this would require unanonymised data, which is not available to us at the moment. Second, explainability needs to be further explored as supply chain risk managers would like to trace why the model is uncertain about a particular link prediction i.e. which evidence were used to produce such output.

References

1. I. Manuj and J. T. Mentzer, "Global supply chain risk management strategies," *International Journal of Physical Distribution & Logistics Management*, vol. 38, pp. 192–223, Apr. 2008.
2. D. Ivanov, "Predicting the impacts of epidemic outbreaks on global supply chains: A simulation-based analysis on the coronavirus outbreak (COVID-19/SARS-CoV-2) case," *Transportation Research. Part E, Logistics and Transportation Review*, vol. 136, p. 101922, Apr. 2020.
3. S. Hosseini and D. Ivanov, "Bayesian networks for supply chain risk, resilience and ripple effect analysis: A literature review," *Expert Systems with Applications*, vol. 161, 2020.
4. J. Hartmann and S. Moeller, "Chain liability in multitier supply chains? Responsibility attributions for unsustainable supplier behavior," *Journal of Operations Management*, vol. 32, pp. 281–294, July 2014.
5. F. Ebinger and B. Omondi, "Leveraging digital approaches for transparency in sustainable supply chains: A conceptual paper," *Sustainability (Switzerland)*, vol. 12, no. 15, 2020.
6. E. Kosasih and A. Brintrup, "Towards Digital Supply Chain Risk Surveillance," May 2022. Accepted: 2022-05-11T23:30:24Z Publisher: Elsevier.
7. P. Wichmann, A. Brintrup, S. Baker, P. Woodall, and D. McFarlane, "Extracting supply chain maps from news articles using deep neural networks," *International Journal of Production Research*, vol. 58, pp. 5320–5336, Sept. 2020.
8. A. Brintrup, P. Wichmann, P. Woodall, D. Mcfarlane, E. Nicks, and W. Krechel, "Predicting Hidden Links in Supply Networks," Jan. 2018.
9. E. E. Kosasih and A. Brintrup, "A machine learning approach for predicting hidden links in supply chain with graph neural networks," *International Journal of Production Research*, vol. 0, pp. 1–14, July 2021. Publisher: Taylor & Francis _eprint: https://doi.org/10.1080/00207543.2021.1956697.
10. A. Aziz, E. E. Kosasih, R.-R. Griffiths, and A. Brintrup, "Data Considerations in Graph Representation Learning for Supply Chain Networks," Tech. Rep. arXiv:2107.10609, International Conference on Machine Learning Workshop on ML4Data, July 2021. arXiv:2107.10609 [cs] type: article.
11. E. E. Kosasih, F. Margaroli, S. Gelli, A. Aziz, N. Wildgoose, and A. Brintrup, "Towards knowledge graph reasoning for supply chain risk management using graph neural networks," *International Journal of Production Research*, Dec. 2022.
12. X. Chen, M. Chen, W. Shi, Y. Sun, and C. Zaniolo, "Embedding uncertain knowledge graphs," pp. 3363–3370, 2019.
13. Q. Wang, Z. Mao, B. Wang, and L. Guo, "Knowledge Graph Embedding: A Survey of Approaches and Applications," *IEEE Transactions on Knowledge and Data Engineering*, vol. 29, pp. 2724–2743, Dec. 2017.
14. A. Bordes, N. Usunier, A. Garcia-Duran, J. Weston, and O. Yakhnenko, "Translating Embeddings for Modeling Multi-relational Data," in *Advances in Neural Information Processing Systems*, vol. 26, Curran Associates, Inc., 2013.
15. T. Trouillon, J. Welbl, S. Riedel, E. Gaussier, and G. Bouchard, "Complex Embeddings for Simple Link Prediction," Tech. Rep. arXiv:1606.06357, arXiv, June 2016. arXiv:1606.06357 [cs, stat] type: article.
16. J. Hu, R. Cheng, Z. Huang, Y. Fang, and S. Luo, "On Embedding Uncertain Graphs," in *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, (Singapore Singapore), pp. 157–166, ACM, Nov. 2017.
17. L. Li, P. Wang, Y. Wang, S. Wang, J. Yan, J. Jiang, B. Tang, C. Wang, and Y. Liu, "A method to learn embedding of a probabilistic medical knowledge graph: Algorithm development," *JMIR Medical Informatics*, vol. 8, no. 5, 2020.
18. X. Chen, M. Boratko, M. Chen, S. S. Dasgupta, X. L. Li, and A. McCallum, "Probabilistic Box Embeddings for Uncertain Knowledge Graph Reasoning," Tech. Rep. arXiv:2104.04597, arXiv, Apr. 2021. arXiv:2104.04597 [cs] type: article.

19. S. Yang and R. Tang, “Learning Knowledge Uncertainty from the Pretrained Language Model,” pp. 37–42, 2021.
20. S. Yang, W. Zhang, and R. Tang, “Fast Confidence Prediction of Uncertainty based on Knowledge Graph Embedding,” 2020.
21. S. Pai and L. Costabello, “Learning Embeddings from Knowledge Graphs With Numeric Edge Attributes,” pp. 2869–2875, 2021. ISSN: 1045-0823.
22. T. N. Kipf and M. Welling, “Semi-Supervised Classification with Graph Convolutional Networks,” Tech. Rep. arXiv:1609.02907, arXiv, Feb. 2017. arXiv:1609.02907 [cs, stat] type: article.
23. W. L. Hamilton, R. Ying, and J. Leskovec, “Inductive Representation Learning on Large Graphs,” Tech. Rep. arXiv:1706.02216, arXiv, Sept. 2018. arXiv:1706.02216 [cs, stat] type: article.
24. C. Morris, M. Ritzert, M. Fey, W. L. Hamilton, J. E. Lenssen, G. Rattan, and M. Grohe, “Weisfeiler and Leman Go Neural: Higher-order Graph Neural Networks,” Tech. Rep. arXiv:1810.02244, arXiv, Nov. 2021. arXiv:1810.02244 [cs, stat] type: article.
25. E. E. Kosasih, J. Cabezas, X. Sumba, P. Bielak, K. Tagowski, K. Idanwekhai, B. A. Tjandra, and A. R. Jamasb, “On graph neural network ensembles for large-scale molecular property prediction,” *CoRR*, vol. abs/2106.15529, 2021.