Meta-Path Learning for Multi-relational Graph Neural Networks

Francesco Ferrini
University of Trento, Italy
francesco.ferrini@unitn.it

Antonio Longa
University of Trento, Italy
antonio.longa@unitn.it

Andrea Passerini
University of Trento, Italy
andrea.passerini@unitn.it

Manfred Jaeger
Aalborg University, Denmark
jaeger@cs.aau.dk

Abstract

Existing multi-relational graph neural networks use one of two strategies for identifying informative relations: either they reduce this problem to low-level weight learning, or they rely on handcrafted chains of relational dependencies, called meta-paths. However, the former approach faces challenges in the presence of many relations (e.g., knowledge graphs), while the latter requires substantial domain expertise to identify relevant meta-paths. In this work we propose a novel approach to learn meta-paths and meta-path GNNs that are highly accurate based on a small number of informative meta-paths. Key element of our approach is a scoring function for measuring the potential informativeness of a relation in the incremental construction of the meta-path. Our experimental evaluation shows that the approach manages to correctly identify relevant meta-paths even with a large number of relations, and substantially outperforms existing multi-relational GNNs on synthetic and real-world experiments.

1 Introduction

Graph neural networks (GNNs) have emerged as a powerful framework for analyzing networked data [6, 8, 18, 24], enabling effective learning and representation of complex relationships in several real-world applications [2, 23, 31, 39]. Standard GNN approaches have mostly focused on homogeneous graphs [7, 30, 34], where all nodes and edges belong to a single type. However, many real-world graph datasets exhibit heterogeneity, with multiple types of nodes and relations [4, 22, 28].

Treating heterogeneous graphs as homogeneous and aggregating information uniformly across all relations is a suboptimal approach, as different relations can convey largely different semantic information about the nodes they connect. A simple and effective strategy to retain the rich semantic information present in heterogeneous graphs is relying on meta-paths, which are chains of relational dependencies (e.g., "actor -> acted in -> movie -> has genre"). The challenge lies in determining the relevant meta-paths in a given graph. Existing methods either rely on predefined meta-paths defined by domain experts [3, 5, 32], which are extremely expensive to collect, or learn "soft" meta-paths by learning to assign weights to relations [25, 37, 38], an approach that only works with few relations and fails to scale to knowledge graphs. Solutions conceived for mining meta-paths from knowledge graphs typically consider relations only, ignoring node features altogether [16, 33].

To overcome these limitations, we propose a novel approach to learn meta-paths and meta-path GNNs that are highly accurate based on a small number of informative meta-paths. Key to our approach is the formalization of a scoring function, inspired by the relational information gain principle [14], that evaluates the potential informativeness of a relation in the incremental construction of the meta-path. This allows to learn a Meta-Path Graph Neural Network (MP-GNN) in which different layers convey information from different relations while retaining node-specific features in the aggregation process.

Francesco Ferrini et al., Meta-Path Learning for Multi-relational Graph Neural Networks. *Proceedings of the Second Learning on Graphs Conference (LoG 2023)*, PMLR 231, Virtual Event, November 27–30, 2023.

The main contributions of this work can be summarized as follows:

- We propose a scoring function evaluating the potential informativeness of a relation in the meta-path construction.
- We introduce MP-GNN, a simple variant of the RGCN architecture, which effectively combines learned meta-paths and node features into a multi-relational graph processing architecture.
- We provide an extensive experimental evaluation on synthetic and real-world datasets showing how our approach substantially outperforms existing multi-relational GNNs when dealing with graphs with a large number of relations.

2 Related work

In recent research, meta-path mining has emerged as an effective approach for analyzing heterogeneous graphs, relying on frequency cutoffs and sequential pattern mining strategies to identify promising meta-paths [17, 26, 36]. In the field of neuro-symbolic reasoning for knowledge graph completion (KGC), various approaches use reinforcement learning-based algorithms to explore relation-paths and derive logical formulas [16, 33]. Other approaches [11, 12, 20], search for the most relevant meta-path using variants of random walk search. A major limitation of all these approaches is that they are incapable of accounting for node features in determining the relevance of a candidate meta-path, making them unusable in knowledge graph embedding scenarios.

In the field of heterogeneous graph embedding, several methods have been proposed to enhance node and graph embedding by incorporating meta-paths. These methods can be broadly categorized into two groups: those using predefined meta-paths and those learning meta-paths by weighting the contribution of different relations.

In the first group, Meta-path Aggregated Graph Neural Network [5] focuses on aggregating node features along predefined meta-paths using an attention mechanism, capturing diverse structural patterns. Heterogeneous Attention Network [32] introduces a hierarchical attention mechanism to handle heterogeneous graphs, enhancing performance and interpretability. GraphMSE [13] tackles the problem of information-rich meta-path selection by aggregating information using different meta-paths and adopting BFS (Breadth First Search) as meta-path expansion strategy. Meta-path extracted graph neural network [3] incorporates message passing and emphasizes interpretability and semantic relationships. However, these approaches require that meta-paths are provided beforehand, something which severely limits their adaptability.

In the second group, Relational Graph Convolutional Networks (RGCN) [25] capture relation-specific patterns with distinct trainable parameters for each edge type. R-HGNN [35] uses a dedicated graph convolution component for unique node representations from relation-specific graphs. RSHN [40] integrates Coarsened Line Graph Neural Network (CL-GNN) for enhanced embedding in large-scale heterogeneous graphs. Graph Transformer Networks (GTN) [37] learn task-specific multi-hop connections (i.e., meta-paths) for useful node representations. FastGTN [38] addresses GTN's high complexity by implicitly transforming graphs. HGN [15] employs GAT as a backbone for a simple yet effective model. HGT [9] uses node and edge-type dependent parameters for heterogeneous attention. MVHRE [19] enriches relational representations for link prediction using a multi-view network representation learning framework. While effective with a small number of candidate relations, these approaches' performance degrades as the number increases, as shown in our experimental evaluation.

3 Preliminary

In this section, we provide an overview of fundamental concepts of our approach.

Heterogeneous graph. A heterogeneous graph is a directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{T}_v, \mathcal{T}_e)$ where $\mathcal{V}$ is the set of nodes or entities and $\mathcal{E}$ is the set of edges. Each node v and edge e has a type, specified by the mapping functions $\tau_v(v) : \mathcal{V} \to \mathcal{T}_v$ and $\tau_e(e) : \mathcal{E} \to \mathcal{T}_e$. Moreover, each node v has a feature vector $x_v \in \mathbb{R}^d$.

Meta-path A meta-path mp is a relation sequence defined on a heterogeneous graph $\mathcal{G}$, denoted in the form $\xrightarrow{r_1}\xrightarrow{r_2}$ ... $\xrightarrow{r_L}$, where $r_1, ..., r_L$ are relation types and for each consecutive pair of relations $\xrightarrow{r_i}\xrightarrow{r_{i+1}}$ the intersection between the valid node types that are the destination of $\xrightarrow{r_i}$ and the valid

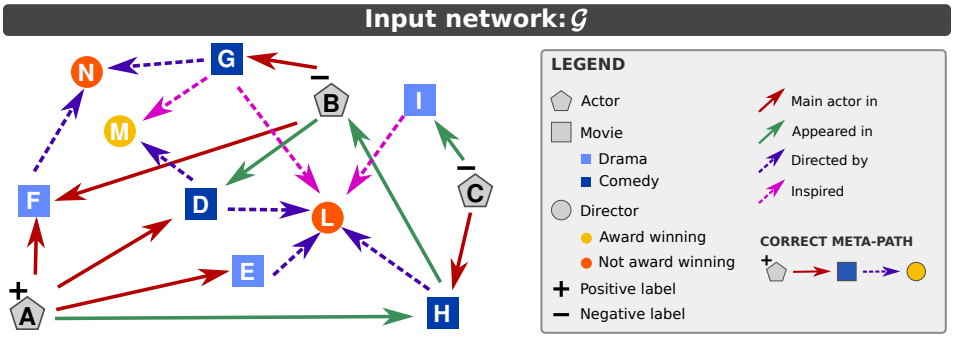


Figure 1: A toy node classification problem. Node shapes indicate types, while node attributes and edge types are encoded as colors. The task consist in labelling actor nodes (pentagons, which do not have attributes). An *Actor* is labelled as positive if involved as main actor in a drama directed by and award winning director.

node types that are the source of $\xrightarrow{r_{i+1}}$ is non-empty. Note that this is a more general definition than the one in [27], in that it allows multiple node types as sources and destinations of a given relation, consistently with what can be found in large general-purpose knowledge graphs.

RGCN layer The relational graph convolutional layer from [25] extends the standard convolution operation on graphs [10] to the multi-relational setting by assigning specific parameters for each relation type. Message passing update for node i at layer l is given by:

$$h_i^{(l+1)} = \sigma\left(W_0^{(l)} h_i^{(l)} + \sum_{r\in\mathcal{R}} \sum_{j\in\mathcal{N}_i^r} \frac{1}{c_{i,r}} W_r^{(l)} h_j^{(l)}\right) \tag{1}$$

where $\mathcal{R}$ is the set of relations in the graph, $\mathcal{N}_i^r$ is the set of r-neighbours of node i and $c_{i,r}$ is a fixed or learnable normalizing parameter.

4 MP-GNN learning

The goal of our approach is to learn relevant meta-paths that can serve as predictive features for the node classification task. [1] Differently from the approaches that use all relations at the same time by weighting each edge type contribution, we focus on finding the relevant chains of relations (i.e., meta-paths) beneficial for making accurate predictions. Note that, differently from what happens in purely relational settings [11, 12, 17, 20, 26, 36], we assume here that the informativeness of a meta-path also depends on the features of the nodes that are being traversed (which include the node type, but also node attributes and potentially pre-computed node embeddings). Our approach accounts for this aspect in mining relevant meta-paths. Meta-paths are constructed in a greedy, incremental approach using the idea of relational information gain [14] to score candidate extensions of an already constructed meta-path prefix. Consider the toy node classification task in Figure 1. To incrementally build the correct meta-path (bottom right in the legend), one has to realize that "Main actor in" is a better candidate than "Appeared in". Intuitively, our scoring function does this by assigning weights (i.e., pseudo-labels) to nodes reached by a candidate relation in such a way that the label of the target node can be inferred by propagating the pseudo-label of the neighbour. Figure 2 shows an example of weight assignment for the "Main actor in" and "Appeared in" relations, indicating a higher score for the former. However, these pseudo-labels only hint at the potential informativeness of the relation. Indeed, being a main actor in a movie is not enough to qualify as an award winning actor, even in the toy example of Figure 1. The movie should be a drama (node feature), and be directed by an award winning director. Whether this potential informativeness actually materializes is determined in the following steps, where the pseudo-labels become new prediction targets for the next extension of the meta-path under construction. Details of this method are described in Section 4.1.

[1] We focused on node classification in this paper, but the approach can be adapted to deal with graph classification tasks, e.g., by considering a supernode connected to all graph nodes as the target.

Once a candidate meta-path has been extracted, it is used to build a MP-GNN in which each layer corresponds to a relation on the meta-path. Section 4.2 presents a formalization of this architecture, and shows how to extend it to account for multiple meta-paths. Finally, these ingredients are combined into an overall algorithm to jointly learn a meta-path and a corresponding MP-GNN. For the sake of readability, the algorithm is presented for the single meta-path case, but its extension to multiple meta-paths using a beam search is straightforward (we employed a beam size equal to three in the experiments). Note that this algorithm is designed to identify existential meta-path features, i.e., cases where the existence of an instance of a meta-path is informative for the class label. Adaptations and extensions where counts or proportions of meta-path realizations are the relevant feature are subject of future work.

4.1 Scoring function

The goal of the scoring function is that of providing a measure of informativeness of a relation towards predicting node labels. We start discussing the first iteration, i.e., identifying the first relation in the meta-path, and then show how the function can be adapted to deal with meta-path extension.

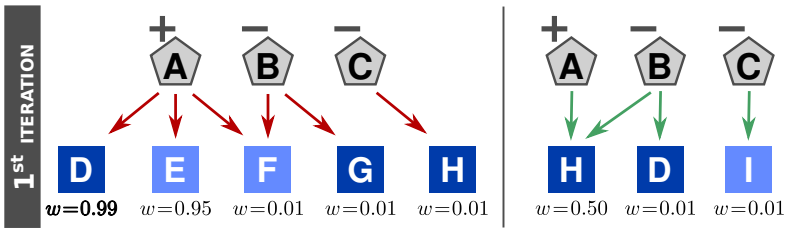


Figure 2: First iteration: the scoring function assigns a high score to the red ("Main actor in") relation (left panel) by giving large weights to movies D and E, that are only connected to a positive node, and small weights to the other movie nodes. On the other hand, the green ("Appeared in") relation (right panel) has low score, as no weight assignment can jointly explain the positive label of the A node and the negative label of the B node.

In the first iteration, the scoring function takes as input a list of nodes together with their target labels. Under the previously introduced existential quantification assumption, a candidate relation r is informative for the label of a node i if at least one of the neighbors $\mathcal{N}_i^r$ of i according to r belongs to the ground-truth meta-path, and i has the right features (remember that the label is assumed to depend on the combination of the meta-path and the features of the nodes being traversed). This can be formalized as follows:

$$\tilde{y}_i^r = \Theta^T h_i^{(0)} \cdot \max_{j \in \mathcal{N}_i^r} w_j \tag{2}$$

Here Θ is a learnable weight vector accounting for the contribution of the node features, while w_j is a learnable node weight that is set (close) to 1 if node j is predicted as belonging to the ground-truth meta-path, and (close to) zero otherwise. The score of r is computed by minimizing the MSE between the predicted and ground truth node labels over Θ and $\mathbf{w}$:

$$s_r = \min_{\Theta, \mathbf{w}} \frac{1}{N} \sum_{i=1}^{N} (\tilde{y}_i^r - y_i)^2 \tag{3}$$

The relation with the minimum score is selected as the first relation of the meta-path.

To explain how the scoring of the following relations in the meta-path works, it is important to remember that the weights $\mathbf{w}$ represent a tentative assignment to neighbours as belonging or not-belonging to the ground-truth meta-path (i.e., their *potential informativeness*). Multiple potential assignments can be minimizers of Eq. 3. In the left panel of Figure 2, where relation r_1 (red) is being scored, one minimizer of Eq. 3 requires $w_E = 1$ (to account for the positive label of node A) and $W_F = 0$ (to account for the negative label of node B). On the other hand, (0,1), (1,0) and (1,1) are all valid assignments to the (W_D, W_E) pair. Indeed, the only constraint that the positive label of A enforces is that the bag (W_D, W_E) contains at least one node with value 1, as happens in multi-instance classification settings [1]. We thus generate labelled bags of nodes for the following iteration(s) of meta-path construction, that will play the role of the node labels y in the initial iteration.

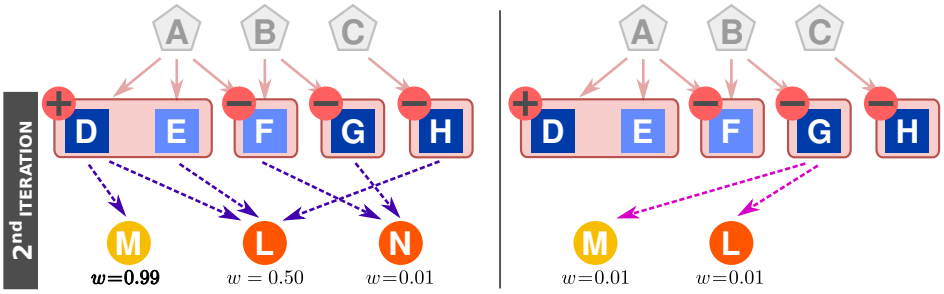


Figure 3: Second iteration: The scoring function assigns a high score to the purple ("Directed by") relation (left panel) by assigning a large weight to the M director, which is only one connected to a positive bag, and small weights to the other directors. On the other hand, the "pink" (Inspired) relation (right panel) gets a low score as no weight assignment is compatible with the positive bag.

Positive bags are computed as follows:

$$B^+(i) = \{j \in \mathcal{N}_i^r \mid \nexists k : j \in \mathcal{N}_k^r \wedge y_k = -1\} \tag{4}$$

where i is a positive-labelled node ($y_i = +1$). Negative bags, on the other hand, are singletons, i.e., given a negatively-labelled node j, we create a negative bag $B^-(k) = \{k\}$ for each of its neighbors $k \in \mathcal{N}_j^r$. The informativeness of the new relation s (as extension of relation r) can now be computed in terms of its potential in predicting bag labels:

$$\tilde{y}^s_{B(i)} = \max_{j \in B(i)} \left(\Theta^T h_j^{(0)} \cdot \max_{k \in \mathcal{N}_j^s} w_k \right) \tag{5}$$

and obtained minimizing MSE at the bag-label level. See Figure 3 for a graphical representation of the components involved.

Once the next relation is selected, the procedure could in principle continue, by further expanding positive bags with a procedure analogous to Eq. 4, where i is itself replaced with a bag of nodes. However, this procedures ends up diluting information too much, so that the informativeness of relations becomes difficult to predict. We rather assign a positive label to a node within a bag if it is used to predict the positive label of the bag (Eq. 2) at least once out of M restarts with randomly initialized weights. See the Appendix for the details.

4.2 MP-GNN

In the single meta-path MP-GNN, a meta-path $mp = r_1, ..., r_L$ induces a multi-relational GNN with L layers, that we denote by MP-GNN(mp). The first layer is associated to the last relation of the meta-path r_L, and so on until the final layer which is associated with r_1. The message passing update is formalized as follows:

$$h_i^{(l+1)} = \sigma \left(W_0^{(l)} h_i^{(l)} + \sum_{j \in \mathcal{N}_i^{r_{L-l+1}}} \frac{1}{|\mathcal{N}_i^{r_{L-l+1}}|} W^{(l)} h_j^{(l)} \right) \tag{6}$$

where l ranges from 1 to L.

The definition can be generalized to deal with multiple meta-paths by concatenating embeddings coming from the different meta-paths:

$$h_i^{(l)} = \Big\|_{k=1}^{K} h_{(i,k)}^{(l)} \tag{7}$$

where K is the number of meta-paths, $h_{(i,k)}^{(l)}$ is the embedding of node i according to meta-path k computed using Eq. 6 and $\|$ is the concatenation operator.

It is worth mentioning here that while this definition of MP-GNN is a straightforward adaptation of the RGCN architecture to deal with learned meta-paths, more complex architectures involving

pre-defined meta-paths could in principle be employed [3, 5, 13, 32]. We opted for this simple choice in the paper so as to best single out the contribution of the scoring function in determining the performance of the resulting architecture.

4.3 Overall algorithm

The overall algorithm for learning MP-GNN is outlined in Algorithm 1. The algorithm takes as inputs a heterogeneous graph $\mathcal{G}$, a set of candidate relations $\mathcal{R}$, a set of node-label pairs $labels$ and a hyper-parameter L_{MAX} indicating the maximal length of candidate meta-paths. The algorithm repeatedly call the scoring function (Eq. 3) to score candidate relations and keeps track of the best scoring one. It then builds an MP-GNN with the current (partial) meta-path and trains it to predict node labels, using F_1 score (computed on a validation set, omitted in the algorithm for the sake of compactness) as final meta-path evaluation metric. Note that this is the only "real" measure of meta-path quality, as the one computed by the scoring function is still a "potential" informativeness, that only fully materializes when the meta-path is embedded into an MP-GNN. The algorithm keeps track of the highest F_1 meta-path prefix so far, and proceeds by generating labelled bags as described in Section 4.1 for the next round of relation scoring.

As previously mentioned, the algorithm is presented for the sake of simplicity in the single meta-path case. However, the actual implementation performs beam search on the space of meta-paths, retaining the K top-scoring ones according to Eq. 3 and concatenating their embeddings into the MP-GNN as per Eq. 7. Notice that in evaluating the resulting MP-GNN, meta-paths not contributing to increasing F_1 are discarded, so as to retain only the informative meta-paths in the final architecture.

Algorithm 1 LEARNMP-GNN algorithm. $\mathcal{G}$ is a heterogeneous graph, $\mathcal{R}$ is the set of possible relations, $labels$ is the initial set of node-label pairs and L_{MAX} is the maximal meta-path length

procedure LEARNMP-GNN($\mathcal{G}, \mathcal{R}, labels, L_{MAX}$)
  Initialize $mp^* \leftarrow [\,]$, $mp \leftarrow [\,]$, $F_1^* \leftarrow 0$, $target \leftarrow labels$
  while $|mp| < L_{MAX}$ **do**
    for $r \in \mathcal{R}$ **do**
      $s_r \leftarrow$ SCORE-RELATION$(\mathcal{G}, target, r)$ ▷ Equation 3
    end for
    $r^* \leftarrow$ best scoring relation
    $mp \leftarrow mp, r^*$
    $gnn \leftarrow$ TRAIN(MP-GNN$(mp), \mathcal{G}, labels)$
    $F_1 \leftarrow$ TEST(gnn)
    if $F_1 > F_1^*$ **then**
      $mp^* \leftarrow mp,\ F_1^* \leftarrow F_1$
    end if
    $target \leftarrow$ GENERATE-BAGS$(target, r^*)$ ▷ Section 4.1
  end while
  return mp^*
end procedure

5 Experimental results

Our experimental evaluation aims to answer the following research questions:

Q1 Can MP-GNN recover the correct meta-path for an increasing number of candidate relations?

Q2 Is MP-GNN competitive with existing approaches in real-world datasets with few relations?

Q3 Does MP-GNN outperform existing approaches in real-world datasets with many relations?

We compared MP-GNN with existing solutions that: 1) do not require to pre-specify relevant meta-paths, 2) can handle (possibly high-dimensional) node features. Given these requirements, we identified the following competitors:

- **RGCN** [25], a generalization of the GCN architecture to the multi-relational case, that employs a different matrix of parameters for each edge type.

- **GTN** [37] can convert an input graph into different meta-path graphs for specific tasks and learn node representations within these graphs.
- **FastGTN** [38], an efficient variant of GTN that avoids adjacency matrices multiplication for graph transformation.
- **R-HGNN** [35], employs a different convolution for each edge type. Finally combines different embeddings with a cross-relation message passing.
- **HGN** [15], utilizes GAT as backbone to design an extremely simple HGNN model.

We implemented MP-GNN using Pytorch Geometric [21], while the code of the competitors was taken from their respective papers. For MP-GNN we used Adam optimizer with a learning rate of 0.01. We set the maximum meta-path length $L_{MAX} = 4$ and the beam size $K = 3$. We used an 80/20/10 split between train, validation and test in all cases, with model selection performed on the validation set for all methods. We employed F1-macro score on the test set as evaluation metric to account for the unbalancing present in many of the datasets. The code is available at LINK.

In the following we report the experimental setting and the results we obtained in addressing each of the research questions under investigation. The statistics of the datasets used in the experiments are reported in the Appendix.

5.1 Q1: MP-GNN consistently identifies the correct meta-path

In order to answer the first research question, we designed a controlled setting where the correct meta-path is known, and experiments can be run for an increasing number of candidate relations. We generated synthetic datasets where nodes are typed A or B, the number of relations $|\mathcal{R}|$ varies in $\{4, 8, 10, 14\}$, and the number of relations that can connect more than one pair of node types (e.g., $A \xrightarrow{r_1} B$ and $A \xrightarrow{r_1} A$). The ground truth meta-path consists of a (valid) sequence of relations and nodes of a given type (e.g., $x \xrightarrow{r_1} A \xrightarrow{r_3} B$, with x being a node of arbitrary type). Nodes are labelled as positive if found to be starting points of a ground-truth meta-path, and negative otherwise. We generated labelled datasets using ground-truth meta-paths of different lenghts $L \in \{2, 3, 4\}$. Details of the different settings can be found in the Appendix (Figure 7).

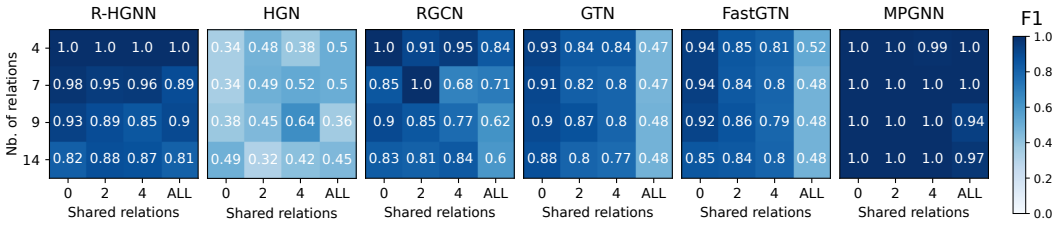


Figure 4: Synthetic setting: F1-score (the darker the better) as a function of the overall number of relations (rows) and the number of shared relations (columns).

Figure 4 shows the F_1 score for each model when varying the overall number of relations and the number of shared relations, for a ground-truth meta-path of length three. Darker cells correspond to higher F_1 value. Results show that the performance of existing multi-relational GNN approaches is severely affected by the relational complexity of the graph, with RGCN and R-HGNN being more sensible to the overall number of candidate relations and GTN and FastGTN having bigger problems with the number of shared relations, while HGN has poor performance in all settings, likely due to its lack of an explicit modelling of relation types. Conversely, MP-GNN consistently achieves optimal or quasi-optimal performance in all settings. Whenever $F_1 = 1$, MP-GNN manages to perfectly recover the ground-truth meta-path, while values smaller than one are due to spurious relations being added at the end of the meta-path (which however have a limited impact on predictive performance).

Figure 5 shows results when increasing the relational complexity of the network *and* the length of the meta-path characterizing the positive class. Each setting corresponds to an entry in the main diagonal of Figure 4, where we additionally varied the length of the meta-path from 2 to 4. Results show that GTN, FastGTN and HGN struggle in most settings, while RGCN and R-HGNN are competitive in the simplest settings (few relations and/or short meta-paths) but its performance quickly degrade when the size of the search space increases. Again, MP-GNN consistently achieves excellent performance in all settings, almost always perfectly recovering the ground-truth meta-path.

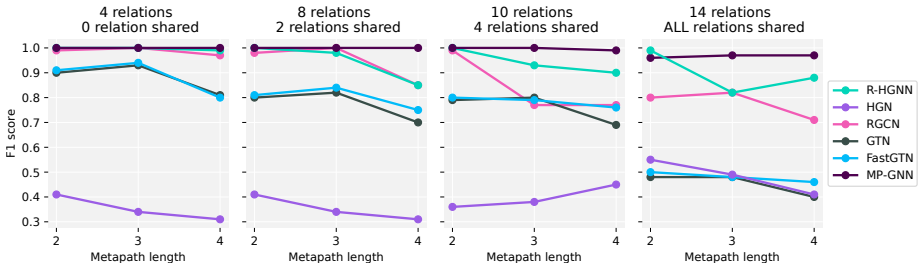


Figure 5: Synthetic setting: F1-score as a function of the ground-truth meta-path length, for an increasing complexity of the search space.

5.2 Q2: MP-GNN achieves state-of-the-art results on real-world datasets with few relations

The second set of experiments focuses on popular real-world benchmarks for multi-relational GNNs. In all cases the task is multi-class classification at the node level. We quickly summarize the characteristics of the benchmarks in the following: **IMDB**: a dataset extracted from the popular Internet Movie Database. It contains 3 types of nodes (movies (M), directors (D) and actors (A)) and uses the genres of movies as labels. **DBLP**: citation network where nodes are of paper (P), author (A) or conference (C) type, connected by edge types PA, AP, PC, CP, and the task is predicting the research area of authors. **ACM**: again a citation network, similar to the one of DBLP with conference nodes replaced by subject (S) nodes (and edge types replaced accordingly).

Table 1: Few-relations datasets. **(Top)**: F_1 scores, mean and std computed over five runs. Best results highlighted in bold. **(Bottom)**: learnt meta-paths for MP-GNN and GTN/FastGTN (which learn the very same meta-paths). Other baselines not reported as they do not explictly extract meta-paths.

Model	DBLP	IMDB	ACM
R-HGNN	0.86(±0.04)	**0.64**(±0.01)	0.9(±0.01)
HGN	**0.94**(±0.01)	0.63(±0.02)	0.92(±0.02)
RGCN	0.91(±0.01)	0.6(±0.01)	0.9(±0.02)
GTN	0.9(±0.01)	0.62(±0.01)	0.91(±0.01)
FastGTN	0.92(±0.00)	0.63(±0.01)	**0.93**(±0.00)
MP-GNN	**0.94**(±0.01)	**0.64**(±0.01)	**0.93**(±0.00)
GTN/ FastGTN	APCPA, APAPA, APA	MAM, MDM, MDMDM	PAP, PSP,
MP-GNN	APCPA, APAPA	MAM, MDM	PAP, PSP

Table 1 (top) shows the F_1 scores achieves by the different methods. As expected, all approaches achieve similar results, which are consistent with the ones observed in previous work [38]. Indeed, the number of relations is very limited (three for IMDB, four for DBLP and ACM) and, most importantly, no relations are shared among different node pair types, substantially restricting the number of candidate meta-paths. Still, MP-GNN achieves slightly better results, most likely thanks to its ability to select a minimal set of meta-paths, as shown in Table 1 (bottom).

5.3 Q3: MP-GNN substantially outperforms competitors on real-world datasets with many relations

The last set of experiments aims to evaluate MP-GNN in a complex real-world setting characterized by a large set of relations, as typical of general-purpose knowledge graphs. We thus designed a set of node-classification tasks over **FB15K-237** [29], which is a large knowledge graph derived from Freebase. Each entity in the graph is associated with a text description, that we transformed into a bag-of-words representation of length 100 (retaining the most frequent words in the dataset). We identified as target labels all many-to-one relations that have from 2 to 20 possible destination types (to avoid having classes with too few examples). Examples include gender, event type and a number of currency-related relations. See the Appendix for the statistics of the datasets.

Table 2 reports F_1 scores for the different methods. GTN and FastGTN have serious difficulties in learning reasonable models in all cases. Indeed, the unbalancing in the class distribution, combined with the large pool of candidate relations to learn from, drives them to boil down to majority class

prediction in most cases. Despite the better performance of RGCN, HGN, and R-HGNN, they still exhibit substantially lower F1-scores compared to MP-GNN. Notably MP-GNN is surpassed only by RGCN and R-HGNN in the "event" and "team sport" classification tasks, respectively. Figure 6 shows some examples of extracted meta-paths for two different classification tasks, namely predicting the currency of domestic tuition fees in educational institutions and predicting the sport a team is playing. In the former case, extracted meta-paths lead to the headquarters of the organization delivering the educational program, which clearly correlate with the currency being used. In the latter case, meta-paths include the league where the team is playing, which again carries information about the sport being played. Note that in both cases, node features are crucial in leveraging meta-path information, as there are not enough examples to generalize via, e.g., specific headquarter or sport league name. Indeed, an ablation experiment excluding node feature information (the typical setting of meta-path mining approaches [11, 17, 26]), shows that none of the methods manages to learn any sensible meta-path, always boiling down to learning majority class prediction rules (see Appendix 8). For the same reasons, plain meta-path mining fails to extract sensible meta-paths, resulting in poor performance (see Appendix 9 for the results using the popular PRA meta-path miner [11]).

Table 2: Many-relations dataset: F1-scores for the different node classification tasks on the FB15K-237 dataset. Results with standard deviations can be found in Table 7 in Appendix. See Table 3 in the Appendix for the meaning of the label acronyms.

Label	R-HGNN	HGN	RGCN	GTN	FastGTN	MP-GNN
PNC	0.72	0.68	0.74	0.33	0.33	**0.83**
EDC	0.6	0.75	0.71	0.12	0.12	**0.96**
EIC	0.63	0.65	0.73	0.12	0.12	**0.8**
ELC	0.47	0.74	0.72	0.12	0.15	**0.78**
FBC	0.45	0.48	0.42	0.14	0.14	**0.61**
GNC	0.8	0.74	0.82	0.19	0.19	**0.90**
OC	0.67	0.73	0.78	0.14	0.14	**0.93**
G	0.81	0.64	0.8	0.44	0.44	**0.84**
TS	**0.67**	0.53	0.62	0.09	0.09	0.63
E	0.89	0.8	**0.98**	0.07	0.07	0.96

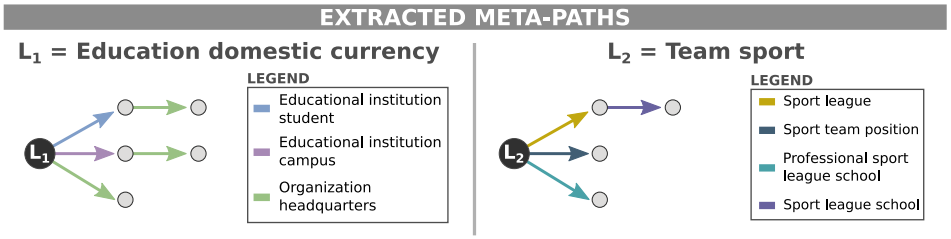


Figure 6: Examples of learned meta-paths on two node classification tasks

Finally, to assess the computational efficiency of MP-GNN, we conducted a running time comparison, detailed in Appendix 10. Results show that our approach is comparable with that of the competitors on the synthetic and few relation (IMDB, DBLP, ACM) datasets. On the freebase tasks, which have a larger set of candidate relations, our approach is more expensive than (most) competitors, but these have substantially lower performance in terms of F1, with the fastest approaches (GTN and FastGTN) completely failing to learn anything sensible.

6 Conclusion

In this work we introduced a novel approach inspired by information theoretic principles to effectively learn meta-paths and meta-path based multi-relational GNNs in settings characterized by a large number of candidate relations combined with arbitrarily rich node features. Our experimental evaluation confirms the potential of the approach in recovering correct (in synthetic tasks) and informative (in real-world tasks) meta-paths despite the large number of candidate relations, a setting where existing multi-relational GNNs struggle to learn meaningful models.

Future work includes generalizing the approach to account for counts or proportions of meta-path realizations as relevant features, as well as more complex relational structures like meta-trees.

Acknowledgments

This research was supported by TAILOR, a project funded by the EU Horizon 2020 research and innovation program under GA No 952215. AL acknowledges the support of the MUR PNRR project FAIR - Future AI Research (PE00000013) funded by the NextGenerationEU.

References

[1] Amores, J.: Multiple instance classification: Review, taxonomy and comparative study. Artificial Intelligence **201**, 81–105 (2013). https://doi.org/https://doi.org/10.1016/j.artint.2013.06.003, https://www.sciencedirect.com/science/article/pii/S0004370213000581 4

[2] Buffelli, D., Vandin, F.: Attention-based deep learning framework for human activity recognition with user adaptation. IEEE Sensors Journal **21**(12), 13474–13483 (2021) 1

[3] Chang, Y., Chen, C., Hu, W., Zheng, Z., Zhou, X., Chen, S.: Megnn: Meta-path extracted graph neural network for heterogeneous graph representation learning. Knowledge-Based Systems **235**, 107611 (2022). https://doi.org/https://doi.org/10.1016/j.knosys.2021.107611, https://www.sciencedirect.com/science/article/pii/S095070512100873X 1, 2, 6

[4] Dong, Y., Chawla, N.V., Swami, A.: Metapath2vec: Scalable representation learning for heterogeneous networks. In: Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. p. 135–144. KDD '17, Association for Computing Machinery, New York, NY, USA (2017). https://doi.org/10.1145/3097983.3098036, https://doi.org/10.1145/3097983.3098036 1

[5] Fu, X., Zhang, J., Meng, Z., King, I.: MAGNN: Metapath aggregated graph neural network for heterogeneous graph embedding. In: Proceedings of The Web Conference 2020. ACM (apr 2020). https://doi.org/10.1145/3366423.3380297 1, 2, 6

[6] Gori, M., Monfardini, G., Scarselli, F.: A new model for learning in graph domains. In: Proceedings. 2005 IEEE International Joint Conference on Neural Networks, 2005. vol. 2, pp. 729–734 vol. 2 (2005). https://doi.org/10.1109/IJCNN.2005.1555942 1

[7] Hamilton, W., Ying, Z., Leskovec, J.: Inductive representation learning on large graphs. Advances in neural information processing systems **30** (2017) 1

[8] Henaff, M., Bruna, J., LeCun, Y.: Deep convolutional networks on graph-structured data (2015) 1

[9] Hu, Z., Dong, Y., Wang, K., Sun, Y.: Heterogeneous graph transformer. In: Proceedings of The Web Conference 2020. p. 2704–2710. WWW '20, Association for Computing Machinery, New York, NY, USA (2020). https://doi.org/10.1145/3366423.3380027, https://doi.org/10.1145/3366423.3380027 2

[10] Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks (2017) 3

[11] Lao, N., Cohen, W.: Relational retrieval using a combination of path-constrained random walks. Machine Learning **81**, 53–67 (10 2010). https://doi.org/10.1007/s10994-010-5205-8 2, 3, 9, 15

[12] Lao, N., Subramanya, A., Pereira, F., Cohen, W.W.: Reading the web with learned syntactic-semantic inference rules. In: Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning. pp. 1017–1026. Association for Computational Linguistics, Jeju Island, Korea (Jul 2012), https://aclanthology.org/D12-1093 2, 3

[13] Li, Y., Jin, Y., Song, G., Zhu, Z., Shi, C., Wang, Y.: Graphmse: Efficient meta-path selection in semantically aligned feature space for graph neural networks. Proceedings of the AAAI Conference on Artificial Intelligence **35**(5), 4206–4214 (May 2021). https://doi.org/10.1609/aaai.v35i5.16544, https://ojs.aaai.org/index.php/AAAI/article/view/16544 2, 6

[14] Lippi, M., Jaeger, M., Frasconi, P., Passerini, A.: Relational information gain. Machine Learning **83**(2), 219–239 (2011). https://doi.org/10.1007/s10994-010-5194-7 1, 3

[15] Lv, Q., Ding, M., Liu, Q., Chen, Y., Feng, W., He, S., Zhou, C., Jiang, J., Dong, Y., Tang, J.: Are we really making much progress? revisiting, benchmarking and refining heterogeneous graph neural networks. In: Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining. p. 1150–1160. KDD '21, Association for Computing Machinery, New York, NY, USA (2021). https://doi.org/10.1145/3447548.3467350, https://doi.org/10.1145/3447548.3467350 2, 7

[16] Meilicke, C., Chekol, M.W., Ruffinelli, D., Stuckenschmidt, H.: Anytime bottom-up rule learning for knowledge graph completion. In: Proceedings of the 28th International Joint Conference on Artificial Intelligence. p. 3137–3143. IJCAI'19, AAAI Press (2019) 1, 2

[17] Meng, C., Cheng, R., Maniu, S., Senellart, P., Zhang, W.: Discovering meta-paths in large heterogeneous information networks. In: Proceedings of the 24th international conference on world wide web. pp. 754–764 (2015) 2, 3, 9

[18] Micheli, A.: Neural network for graphs: A contextual constructive approach. IEEE Transactions on Neural Networks **20**(3), 498–511 (2009). https://doi.org/10.1109/TNN.2008.2010350 1

[19] Mitra, A., Vijayan, P., Singh, S.R., Goswami, D., Parthasarathy, S., Ravindran, B.: Revisiting link prediction on heterogeneous graphs with a multi-view perspective. 2022 IEEE International Conference on Data Mining (ICDM) pp. 358–367 (2022), https://api.semanticscholar.org/CorpusID:256463320 2

[20] Neelakantan, A., Roth, B., McCallum, A.: Compositional vector space models for knowledge base completion. In: Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers). pp. 156–166. Association for Computational Linguistics, Beijing, China (Jul 2015). https://doi.org/10.3115/v1/P15-1016, https://aclanthology.org/P15-1016 2, 3

[21] Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Köpf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., Chintala, S.: Pytorch: An imperative style, high-performance deep learning library (2019) 7

[22] Ringler, D., Paulheim, H.: One knowledge graph to rule them all? analyzing the differences between dbpedia, yago, wikidata & co. In: KI 2017: Advances in Artificial Intelligence: 40th Annual German Conference on AI, Dortmund, Germany, September 25–29, 2017, Proceedings 40. pp. 366–372. Springer (2017) 1

[23] Samoaa, H.P., Longa, A., Mohamad, M., Chehreghani, M.H., Leitner, P.: Tep-gnn: Accurate execution time prediction of functional tests using graph neural networks. In: Product-Focused Software Process Improvement: 23rd International Conference, PROFES 2022, Jyväskylä, Finland, November 21–23, 2022, Proceedings. pp. 464–479. Springer (2022) 1

[24] Scarselli, F., Gori, M., Tsoi, A.C., Hagenbuchner, M., Monfardini, G.: The graph neural network model. IEEE transactions on neural networks **20**(1), 61–80 (2008) 1

[25] Schlichtkrull, M., Kipf, T.N., Bloem, P., van den Berg, R., Titov, I., Welling, M.: Modeling relational data with graph convolutional networks (2017) 1, 2, 3, 6

[26] Shi, B., Weninger, T.: Mining interesting meta-paths from complex heterogeneous information networks. In: 2014 IEEE International Conference on Data Mining Workshop. pp. 488–495 (2014). https://doi.org/10.1109/ICDMW.2014.25 2, 3, 9

[27] Sun, Y., Han, J., Yan, X., Yu, P.S., Wu, T.: Pathsim: Meta path-based top-k similarity search in heterogeneous information networks. Proc. VLDB Endow. **4**(11), 992–1003 (aug 2011). https://doi.org/10.14778/3402707.3402736, https://doi.org/10.14778/3402707.3402736 3

[28] Sun, Z., Hu, W., Li, C.: Cross-lingual entity alignment via joint attribute-preserving embedding (2017) 1

[29] Toutanova, K., Chen, D.: Observed versus latent features for knowledge base and text inference. In: Proceedings of the 3rd Workshop on Continuous Vector Space Models and their Compositionality. pp. 57–66. Association for Computational Linguistics, Beijing, China (Jul 2015). https://doi.org/10.18653/v1/W15-4007, https://aclanthology.org/W15-4007 8

[30] Veličković, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., Bengio, Y.: Graph attention networks. arXiv preprint arXiv:1710.10903 (2017) 1

[31] Viñas, R., Joshi, C.K., Georgiev, D., Dumitrascu, B., Gamazon, E.R., Liò, P.: Hypergraph factorisation for multi-tissue gene expression imputation. bioRxiv pp. 2022–07 (2022) 1

[32] Wang, X., Ji, H., Shi, C., Wang, B., Cui, P., Yu, P., Ye, Y.: Heterogeneous graph attention network (2021) 1, 2, 6

[33] Xiong, W., Hoang, T., Wang, W.Y.: Deeppath: A reinforcement learning method for knowledge graph reasoning (2018) 1, 2

[34] Xu, K., Hu, W., Leskovec, J., Jegelka, S.: How powerful are graph neural networks? arXiv preprint arXiv:1810.00826 (2018) 1

[35] Yu, L., Sun, L., Du, B., Liu, C., Lv, W., Xiong, H.: Heterogeneous graph representation learning with relation awareness. CoRR **abs/2105.11122** (2021), https://arxiv.org/abs/2105.11122 2, 7

[36] Yu, X., Sun, Y., Norick, B., Mao, T., Han, J.: User guided entity similarity search using meta-path selection in heterogeneous information networks. Proceedings of the 21st ACM international conference on Information and knowledge management (2012), https://api.semanticscholar.org/CorpusID:14239375 2, 3

[37] Yun, S., Jeong, M., Kim, R., Kang, J., Kim, H.J.: Graph transformer networks. In: Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., Garnett, R. (eds.) Advances in Neural Information Processing Systems. vol. 32. Curran Associates, Inc. (2019), https://proceedings.neurips.cc/paper_files/paper/2019/file/9d63484abb477c97640154d40595a3bb-Paper.pdf 1, 2, 7

[38] Yun, S., Jeong, M., Yoo, S., Lee, S., Yi, S.S., Kim, R., Kang, J., Kim, H.J.: Graph transformer networks: Learning meta-path graphs to improve gnns. Neural Networks **153**, 104–119 (2022). https://doi.org/https://doi.org/10.1016/j.neunet.2022.05.026, https://www.sciencedirect.com/science/article/pii/S0893608022002003 1, 2, 7, 8

[39] Zhang, X.M., Liang, L., Liu, L., Tang, M.J.: Graph neural networks and their current applications in bioinformatics. Frontiers in genetics **12**, 690049 (2021) 1

[40] Zhu, S., Zhou, C., Pan, S., Zhu, X., Wang, B.: Relation structure-aware heterogeneous graph neural network. In: 2019 IEEE International Conference on Data Mining (ICDM). pp. 1534–1539 (2019). https://doi.org/10.1109/ICDM.2019.00203 2

7 Appendix

7.1 Weight initialization for the scoring function

Given a relation r to be evaluated for potential informativeness, node weights are initialized with values that are consistent with the assumption that r determines the target values to be predicted. As discussed in Section 4.1, if a node has a negative label, its neighbours according to relation r cannot be part of a ground-truth meta-path, thus their weight should be close to zero. On the other hand, if all the nodes that are connected to a given node via relation r have a positive label, than that given node could be part of a ground-truth meta-path. We thus initialized node weights are follows:

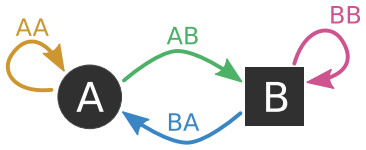
$$w[i] = \min_{j\,:\,i \in \mathcal{N}_j^r} L[j] + \epsilon \tag{8}$$

where $j\ :\ i \in \mathcal{N}_j^r$ states that node j is connected to i via relation r, $L[j]$ is the label of node j and $\epsilon \in [-0.3, 0.3]$ is a random value that helps breaking up symmetries during the weight learning process.

7.2 Re-label nodes inside bags

As discussed in Section 4.1, from the second iteration onwards targets are associated to bags rather than individual nodes. Adapting the procedure in Eq 4 replacing node i with a bag of nodes ends up diluting information too much, so that the informativeness of relations becomes difficult to predict. On the other hand, directly using the weights learned using 3 typically leads to an underestimation of

the set of positive nodes, as it is sufficient that one node in the set of neighbours of a positive node i has a large weight for Equation 2 to compute the correct label for i. Empirically, we found that running the weight learning procedure M times, with different random initializations of weights as discussed in Section 7.1, and retaining for each node its maximum weight across runs, strikes a better balance between precision and coverage in positive node prediction, and thus a better estimate of the informativeness of the corresponding relation. We set $M = 10$ in all our experiments.



Shared relations

Nb. Relations	0	2	4	ALL
4	AA = 0 AB = 1 BA = 2 BB = 3	AA = 0,1 AB = 1 BA = 2,3 BB = 2	AA = 0,3 AB = 1,2 BA = 2,3 BB = 0,1	AA = 0-3 AB = 0-3 BA = 0-3 BB = 0-3
8	AA = 0,1 AB = 2,3 BA = 4,5 BB = 6,7	AA = 0,7 AB = 1,2 BA = 2,3,5 BB = 3,4	AA = 0,1,5 AB = 1,2,7 BA = 4,5,6 BB = 0,7	AA = 0-7 AB = 0-7 BA = 0-7 BB = 0-7
10	AA = 0,1,2 AB = 3,4 BA = 5,6,7 BB = 8,9	AA = 0,1,2 AB = 0,3,4 BA = 5,6,7 BB = 2,8,9	AA = 0,1,2,7 AB = 0,3,4 BA = 5,6,7 BB = 2,3,8,9	AA = 0-9 AB = 0-9 BA = 0-9 BB = 0-9
14	AA = 0,1,2 AB = 3,4,5 BA = 6,7,8,9 BB = 10,11,12,13	AA = 0,1,2,9 AB = 3,4,5,10 BA = 6,7,8,9 BB = 10,11,12,13	AA = 0,1,2,8,9 AB = 3,4,5,10 BA = 6,7,8,9,11 BB = 10,11,12,13	AA = 0-13 AB = 0-13 BA = 0-13 BB = 0-13

Figure 7: Synthetic experimental setting. Each cell is a different experimental setting. A and B are different node types, while $AB = 2, 3$ indicates a setting in which A nodes can be connected to B nodes via relation 2 or 3. Moving from top to bottom we have settings with a large set of relations, while moving from left to right we have settings where relations are progressively less "pure", i.e., the same relation can connect different node-type pairs (e.g., in the second column of the first row there are 2 shared relations: relation 1 that can connect A to A but also A to B, and relation 2 that can connect B to A but also B to B). The last column indicates settings in which any relation is valid between any node-pair type.

Table 3: Real-world datasets statistics. PNC: Person Net Currency (currency associated with an individual's compensation), EDC: Education Domestic Currency (currency associated with domestic tuition fees in educational institutions), EIC: Education International Currency (currency associated with international tuition fees in educational institutions), ELC: Education Local Currency (currency associated with local tuition fees in educational institutions), FBC: Film Budget Currency (currency associated with film budget), GNC: Gdp Nominal Currency (currency associated with GDP nominal value), OC: Organization Currency (currency associated with organization), G: Gender (gender of a person), TS: Team Sport (sport of a specific team), E: Event (links recurring events to their repetitions).

Dataset	# Nodes	# Relations	# Edges	# Features	# Classes	# Target nodes
DBLP	18405	4	67496	334	4	4057
ACM	8994	4	25922	1902	3	3025
IMDB	12772	3	37288	1256	3	2939
FB15K-237 (PNC)	14951	236	306711	100	3	583
FB15K-237 (EDC)	14951	236	308201	100	6	481
FB15K-237 (EIC)	14951	236	305559	100	6	119
FB15K-237 (ELC)	14951	236	307017	100	6	361
FB15K-237 (FBC)	14951	236	315131	100	7	1471
FB15K-237 (GNC)	14951	236	305827	100	3	194
FB15K-237 (OC)	14951	236	307611	100	6	460
FB15K-237 (G)	14951	236	305557	100	2	4530
FB15K-237 (TS)	14951	236	453554	100	9	523
FB15K-237 (E)	14951	236	310742	100	9	134

8 Learning meta-path without node features

In this section, we extract meta-paths using the scoring function, excluding the employment of node features. Following this, we apply the identified meta-paths to train MPGNN. A brief analysis of Table 4 clearly illustrates the crucial impact of node features in extracting valuable meta-paths. The only exception is observed in the case of synthetic dataset 1, which is unsurprising given its inherent simplicity.

Table 4: MPGNN with and without considering node features in the scoring function.

Dataset	With NF	Without NF
Synthetic 1	**1**($\pm$0.00)	**1**($\pm$0.00)
Synthetic 2	**1**($\pm$0.00)	0.79($\pm$0.05)
Synthetic 3	**1**($\pm$0.00)	0.48($\pm$0.03)
Synthetic 4	**0.97**($\pm$0.02)	0.49($\pm$0.05)
IMDB	**0.64**($\pm$0.01)	0.55($\pm$0.03)
DBLP	**0.94**($\pm$0.01)	0.79($\pm$0.02)
ACM	**0.93**($\pm$0.00)	0.83($\pm$0.03)
PNC (FB15K)	**0.83**($\pm$0.01)	0.49($\pm$0.03)
EDC (FB15K)	**0.96**($\pm$0.01)	0.55($\pm$0.04)
TS (FB15K)	**0.63**($\pm$0.03)	0.52($\pm$0.01)

9 Comparison with meta-path miner

The scoring function we propose incrementally builds relevant meta-paths. In this section, we replace the meta-paths identified by the scoring function with those obtained through conventional mining methods. In particular, we rank meta-paths using the well-known path-ranking-algorithm (PRA)[11]. Subsequently, we choose the top three most relevant meta-paths with lengths of 2, 3, and 4. These selected meta-paths are then used to train the MP-GNN. In Table 5, you can observe the macro-F1 scores for three distinct Freebase datasets: PNC, EDC, and TS. It's important to note that we did not conduct this experiment with IMBD, DBLP, and ACM due to their limited number of relations (3 and 4). Table 5 clearly indicates that the meta-paths identified by the scoring function exhibit significantly higher predictive capabilities compared to those obtained through PRA.

Table 5: Macro-F1 scores for three distinct Freebase datasets

		PNC	EDC	TS
PRA + MP-GNN	meta-path length = 2	0.49	0.14	0.20
	meta-path length = 3	0.49	0.14	0.11
	meta-path length = 4	0.49	0.14	0.11
Scoring Function + MP-GNN		**0.83**	**0.96**	**0.63**

10 Execution Time

In Table 6, we provide information on the training time for each model on individual datasets, along with the corresponding F1 scores enclosed in parentheses. Additionally, we present the average execution time and F1 score across the three benchmark datasets (IMBD, DBLP, and ACM), as well as for Freebase and synthetic datasets. All the models achieve a similar average accuracy on the three benchmark datasets, with RGCN displaying the shortest execution time and GTN the longest. Despite being the second slowest model in terms of execution time, our model achieves the highest average F1 score. It is important to emphasize that our model is designed to learn meaningful meta-paths in networks with a multitude of relation types, whereas the three benchmark datasets only have a limited number of relation types. In the Freebase network, both FastGTN and GTN boast the shortest execution times, yet their F1 scores are significantly lower, rendering them ineligible for further consideration. Despite Simple-HGN being the quickest model, its F1 score falls significantly short, averaging 15 points lower than MP-GNN. On the other hand, RGCN is the second fastest model, but

once more, its F1 score lags behind our approach by an average of 9 points. Lastly, concerning the synthetic datasets, our model ranks third in terms of execution time, with Simple-HGN excluded from the analysis due to its poor F1 scores. It's worth highlighting that the fastest model, FastGTN, and the second fastest, RGCN, both lag behind MP-GNN by 23 and 17 points, respectively. In general, our approach is consistently neither the slowest nor the fastest; however, it consistently attains the highest average F1 score in all scenarios.

Table 6: Execution time in seconds, with the F1 score in parentheses.

	R-HGNN	Simple-HGN	RGCN	GTN	FastGTN	MP-GNN
IMDB	680 (0.64)	740 (0.63)	650 (0.60)	1500 (0.62)	910 (0.63)	1000 (0.64)
DBLP	720 (0.86)	870 (0.94)	780 (0.91)	1630 (0.90)	990 (0.92)	1180 (0.94)
ACM	940 (0.90)	750 (0.92)	870 (0.90)	1420 (0.91)	870 (0.93)	960 (0.93)
Mean	780 (0.80)	786 (0.83)	767 (0.80)	1517 (0.81)	923 (0.83)	1046 (0.84)
PNC	7230 (0.72)	560 (0.68)	1830 (0.74)	180 (0.33)	150 (0.33)	2870 (0.83)
EDC	7356 (0.60)	670 (0.75)	1540 (0.71)	190 (0.12)	130 (0.12)	3220 (0.96)
EIC	7020 (0.63)	460 (0.65)	2040 (0.73)	180 (0.12)	110 (0.12)	2480 (0.80)
ELC	820 (0.47)	760 (0.74)	1380 (0.72)	180 (0.15)	130 (0.15)	3010 (0.78)
FBC	5900 (0.45)	410 (0.48)	1190 (0.42)	190 (0.14)	100 (0.14)	2880 (0.60)
GNC	8230 (0.80)	670 (0.74)	1680 (0.82)	175 (0.19)	100 (0.19)	3220 (0.90)
OC	3790 (0.67)	670 (0.73)	1970 (0.78)	185 (0.14)	120 (0.14)	2980 (0.93)
G	5980 (0.81)	450 (0.64)	2010 (0.80)	140 (0.44)	120 (0.44)	2990 (0.84)
TS	5000 (0.67)	410 (0.53)	1995 (0.62)	200 (0.09)	120 (0.09)	3155 (0.63)
E	6790 (0.89)	690 (0.80)	2005 (0.98)	170 (0.07)	140 (0.07)	3040 (0.96)
Mean	5812 (0.67)	575 (0.67)	1764 (0.73)	179 (0.18)	122 (0.18)	2984 (0.82)
Synt 1	320 (1.00)	50 (034)	200 (1.00)	230 (0.93)	210 (0.94)	245 (1.00)
Synt 2	310 (1.00)	66 (0.48)	240 (0.91)	210 (0.84)	180 (0.85)	300 (1.00)
Synt 3	350 (1.00)	68 (0.38)	300 (0.95)	310 (0.84)	280 (0.81)	345 (0.99)
Synt 4	410 (1.00)	78 (0.50)	390 (0.84)	480 (0.47)	320 (0.52)	430 (1.00)
Synt 5	450 (0.98)	67 (0.34)	400 (0.85)	400 (0.91)	360 (0.94)	380 (1.00)
Synt 6	430 (0.95)	120 (0.49)	390 (1.00)	460 (0.82)	400 (0.84)	450 (1.00)
Synt 7	450 (0.96)	94 (0.52)	450 (0.68)	500 (0.8)	490 (0.8)	480 (1.00)
Synt 8	520 (0.89)	128 (0.50)	460 (0.71)	720 (0.47)	450 (0.48)	440 (1.00)
Synt 9	560 (0.93)	135 (0.38)	425 (0.90)	530 (0.90)	430 (0.92)	510 (1.00)
Synt 10	590 (0.89)	90 (0.45)	580 (0.85)	600 (0.87)	520 (0.86)	540 (1.00)
Synt 11	630 (0.85)	200 (0.64)	495 (0.77)	640 (0.8)	510 (0.79)	500 (0.94)
Synt 12	630 (0.90)	130 (0.36)	500 (0.62)	605 (0.48)	525 (0.48)	560 (1.00)
Synt 13	610 (0.82)	175 (0.49)	565 (0.83)	670 (0.88)	570 (0.85)	580 (1.00)
Synt 14	650 (0.88)	200 (0.32)	595 (0.81)	650 (0.8)	600 (0.84)	560 (1.00)
Synt 15	600 (0.87)	185 (0.42)	650 (0.84)	710 (0.77)	680 (0.8)	590 (1.00)
Synt 16	660 (0.81)	200 (0.45)	700 (0.6)	720 (0.48)	705 (0.48)	605 (0.97)
Mean	523 (0.92)	129 (0.44)	476 (0.82)	547 (0.75)	468 (0.76)	484 (0.99)

11 FB15K-237 Results with standard deviations

Table 7: Many-relations dataset: F1-scores for the different node classification tasks on the FB15K-237 dataset. Mean and standard deviation computed over five runs with different random seeds. Best results are highlighted in bold. See Table 3 in the Appendix for the meaning of the label acronyms.

Label	R-HGNN	HGN	RGCN	GTN	FastGTN	MP-GNN
PNC	0.72(±0.04)	0.68(±0.05)	0.74(±0.03)	0.33(±0.00)	0.33(±0.00)	**0.83**(±0.01)
EDC	0.6(±0.05)	0.75(±0.02)	0.71(±0.02)	0.12(±0.00)	0.12(±0.00)	**0.96** (±0.01)
EIC	0.63(±0.06)	0.65(±0.04)	0.73(±0.01)	0.12(±0.00)	0.12(±0.00)	**0.8**(±0.03)
ELC	0.47(±0.07)	0.74(±0.01)	0.72(±0.02)	0.12(±0.00)	0.15(±0.00)	**0.78**(±0.02)
FBC	0.45(±0.04)	0.48(±0.02)	0.42(±0.00)	0.14(±0.00)	0.14(±0.00)	**0.61**(±0.01)
GNC	0.8(±0.05)	0.74(±0.02)	0.82(±0.03)	0.19(±0.00)	0.19(±0.00)	**0.9**(±0.00)
OC	0.67(±0.02)	0.73(±0.04)	0.78(±0.01)	0.14(±0.00)	0.14(±0.00)	**0.93**(±0.01)
G	0.81(±0.01)	0.64 (±0.01)	0.8(±0.01)	0.44(±0.00)	0.44(±0.00)	**0.84**(±0.04)
TS	**0.67**(±0.04)	0.53(±0.04)	0.62(±0.01)	0.09(±0.00)	0.09(±0.00)	0.63(±0.03)
E	0.89(±0.02)	0.8(±0.03)	**0.98**(±0.00)	0.07(±0.00)	0.07(±0.00)	0.96(±0.00)