Dynamically Pruned Message Passing Networks for Large-Scale Knowledge Graph Reasoning

Xiaoran Xu
Hulu LLC
xiaoran.xu@hulu.com

Wei Feng
Hulu LLC
wei.feng@hulu.com

Yunsheng Jiang
Hulu LLC
yunsheng.jiang@hulu.com

Xiaohui Xie
Hulu LLC
xiaohui.xie@hulu.com

Zhiqing Sun
Carnegie Mellon University
zhiqings@andrew.cmu.edu

Zhi-Hong Deng
Peking University
zhdeng@pku.edu.cn

Abstract

We propose *Dynamically Pruned Message Passing Networks* (DPMPN) for large-scale knowledge graph reasoning. In contrast to existing models, embedding-based or path-based, we learn an input-dependent subgraph to explicitly model a sequential reasoning process. Each subgraph is dynamically constructed, expanding itself selectively under a flow-style attention mechanism. In this way, we can not only construct graphical explanations to interpret prediction, but also prune message passing in Graph Neural Networks (GNNs) to scale with the size of graphs. We take the inspiration from the consciousness prior proposed by Bengio [1] to design a two-GNN framework to encode global input-invariant graph-structured representation and learn local input-dependent one coordinated by an attention module. Experiments show the reasoning capability of our model which is to provide clear graphical explanations as well as to predict results accurately, outperforming most state-of-the-art methods in knowledge base completion tasks.

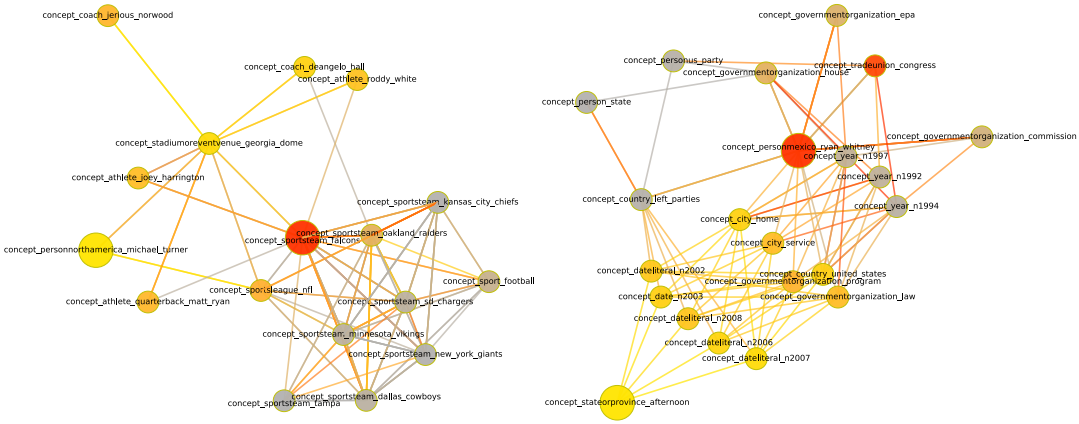
1 Introduction

Modern deep learning systems need to acquire the reasoning capability beyond their black-box nature to produce interpretable predictions [2, 3]. In what form we can better model a reasoning process should be given more thought than just caring about a final correct prediction. Intuitively, a reasoning process can be regarded as a sequence of using existing facts to establish new knowledge, step by step, and finally drawing conclusions in the form of constructing explanations as well as making predictions. Therefore, it needs an explicit modeling to identify and organize reasoning steps to form a clear interpretable representation during predicting. A natural idea is to use graph-structured representation where a semantic unit or pairwise relation can be explicitly represented by a node or an edge as building blocks to support graph-based reasoning, a more flexible form in contrast to rigid deductive logical reasoning [4, 5].

Graph-based reasoning can be applied to a wide variety of real-world scenarios. Here, we choose knowledge graph-related tasks to explore due to its representativeness. In knowledge base completion (KBC) tasks, embedding-based models [6, 7, 8, 9, 10, 11] can easily obtain a very competitive score by fitting data using various neural network techniques, but lacking an explicit modeling on constructing explanations prevents it from being interpretable, a critical property of reasoning, since Euclidean embedding space will not produce a clearly stated and human-readable representation.

Recent work on knowledge graph (KG) reasoning focuses on path-based [12, 13, 14, 15, 16, 17] or logic-like models [18, 19]. Most of them construct an explicit path to model an iterative decision-

Preprint. Under review.



(a) The *AthletePlaysForTeam* task.

(b) The *OrganizationHiredPerson* task.

Figure 1: Subgraph visualization of reasoning results on two examples from NELL995's test data. One is for the *AthletePlaysForTeam* task and the other for the *OrganizationHiredPerson* task. Each task has a graph with ten thousands of nodes and edges. The big yellow in each part represents a given head and the big red represents a predicted tail. Color on the rest indicates attention scores over a T-step reasoning process, where grey means less attention, yellow means more attention gained during early steps, and red means gaining more attention when getting closer to the final step.

making process using reinforcement learning and recurrent networks. However, a question is: do we have a better form, more flexible and interpretable, to express reasoning in the graph context rather than one or several paths. To this end, we propose to learn a subgraph starting from a head node and expanding itself conditionally and selectively according to a query relation, where a tail node is predicted after the final expansion. To better explain how the tail is determined through a series of expansions, we weigh, prune and save intermediate nodes selected, step by step, to capture long-range dependence and yield a concise and compact subgraph explanation for tail prediction as shown in Figure 1.

Graph reasoning can be powered by Graph Neural Networks. Graph reasoning demands a way to effectively learn about entities, relations, and rules of composing them, that is, an ability to achieve combinatorial generalization by manipulating structured knowledge and producing structured explanations. Graph Neural Networks (GNNs) provide such structured representation and structured computation, and also inherit powerful data-fitting capacity from deep neural networks [20, 4]. Specifically, GNNs follow a neighborhood aggregation scheme, recursively aggregating and transforming neighboring nodes' representations to update each node's representations. Therefore, after T iterations of aggregation, each node can carry the structured information within the node's T-hop neighborhood [21, 22].

GNNs need graphical attention expression to interpret. Neighborhood attention operation is a popular way to implement attention mechanism on graphs [23, 24] by using multi-head self-attention to focus on specific interactions with neighbors when aggregating messages. However, we argue that a way of graphical attention expression should be considered not only to facilitate more flexible structured computation but also to construct dynamically pruned structured explanations. We present three considerations: (1) selecting nodes based on currently operated subgraphs, that is, first attending over nodes inside a subgraph to pick a smaller core set and then attending over the neighborhood of this picked core set to expand the subgraph, (2) breaking isolation of attention operations used each step and propagating attention across steps, a way to perform a discrete version of normalizing flow [25] over nodes, to capture long-range statistical dependence, and (3) that such flow-style attention mechanism should model a changing node probability distribution, that is, a Markov process driven by step-varying transition matrices. Besides, we should use an attention module disentangled from the pipeline of aggregating and transforming representations in GNNs to explicitly model a reasoning process on graphs out of low-level representation computation.

GNNs need input-dependent pruning to scale. GNNs are notorious for its poor scalability due to its heavy computation complexity. Consider, for example, one message passing iteration performed over a graph with $|V|$ nodes and $|E|$ edges. It has quadratic complexity in the number of nodes, $\mathcal{O}(|V|^2)$, if the graph is fully connected. Even if the graph is sparse so that the complexity can be

reduced to $\mathcal{O}(|E|)$ by exploiting structural sparsity, it is still problematic when meeting large graphs with millions of nodes and edges. Besides, mini-batch based training with batch size B and high dimensions D would make things worse, leading to the complexity of $\mathcal{O}(BD|E|)$. We argue that this situation can be avoided by learning input-dependent pruning, as in most cases an input example only uses a small fraction of the entire graph, and it is wasteful to perform homogeneous graph computation on the entire graph for each input. Therefore, we propose to prune message passing depending on specific inputs so that message passing runs on dynamical computation graphs instead of the static one.

Cognitive intuition of the consciousness prior. The notion of attentive awareness has been shared by cognitive science communities in several theories [26, 27]. Bengio [1] brought this notion into deep learning models in his *consciousness prior* proposal. He pointed out a process of disentangling high-level abstract factors from full underlying representation to form a low-dimensional combination of a few selected factors to constitute a conscious thought, and emphasized the role of attention in expressing awareness during this process. Bengio proposed to use two recurrent neural networks (RNNs) to encode two types of state: the unconscious state represented by a full high-dimensional vector before applying attention, and the conscious state by a derived low-dimensional vector after applying attention.

In our work, we use two GNNs to encode such two types of state into node representation vectors. Note that standard message passing runs globally so that messages gathered at one node may come from everywhere and get further entangled after applying aggregation operations. Therefore, we consider drawing an input-dependent or context-aware local subgraph to constrain message passing. Besides, we hope to be able to access certain global information about the graph structure to get a boarder view even though we will focus on a local subgraph per input. Inspired by the consciousness prior [1], we apply a novel flow-style graphical attention mechanism to connect and organize these two GNNs. The bottom GNN performs input-invariant standard message passing globally, called **Inattentive GNN (IGNN)**, and the above GNN performs input-dependent pruned message passing locally, called **Attentive GNN (AGNN)**. The intuition is that IGNN offers a pool of raw representations for AGNN to select from, entangled but rich, while AGNN focuses on multiple input-dependent subgraphs consisting of a few selected nodes and their edges, cohesive with sharp semantics, disentangled from the full graph. Nodes inside such a subgraph should be more densely connected to form a small community to further exchange information and make decisions collectively on how to grow the subgraph next. In experiments, we find our model can run on very large graphs with millions of edges, such as the YAGO3-10 dataset, even just using a laptop without causing the out-of-memory error. Our prediction results on KBC tasks attain highly competitive scores on HITS@1,3 and the mean reciprocal rank (MRR) compared to the best embedding-based method so far[1]. Besides, we provide interpretations which they do not have.

2 Problem Formulation

Notation. We use a supervised setting with training data $\{(x_i, y_i)\}_{i=1}^N$ where x_i is an input and y_i is a target. We denote a full graph by $\mathcal{G}$ with node set $\mathcal{V}$ and edge set $\mathcal{E}$, and denote an input-dependent subgraph by $G(x)$ with node set $V_{G(x)}$ and edge set $E_{G(x)}$. We also denote the set of edge types (or relation types) by $\mathcal{R}$. We require each subgraph G to hold $E_G = \{(v, u) \in \mathcal{E} : v, u \in V_G\}$, so that we can define $G = G_1 \cup G_2$ if $V_G = V_{G_1} \cup V_{G_2}$ and define $G_1 \subseteq G_2$ if $V_{G_1} \subseteq V_{G_2}$. We define the boundary of a subgraph as ∂G if $V_{\partial G} = N(V_G) - V_G$ where $N(V_G)$ means the union of neighbors of all the nodes in V_G. We also define high-order boundaries such as $\partial^2 G$ if $V_{\partial^2 G} = N(N(V_G)) \cup N(V_G) - V_G$. Trainable parameters include node embeddings $\{\boldsymbol{e}_v\}_{v\in\mathcal{V}}$, relation type embeddings $\{\boldsymbol{e}_r\}_{r\in\mathcal{R}}$, and neural network weights used in two GNNs and an attention module. When performing standard or pruned message passing, node embeddings and relation type embeddings will be indexed according to the operated graph, and thus we denote them by $\boldsymbol{\theta}_{\mathcal{G}}$ or $\boldsymbol{\theta}_{G(x)}$. We denote batch size by B and dimensions by D. For IGNN, we use $\boldsymbol{\mathcal{H}}^t$ of size $|\mathcal{V}| \times D$ to denote node hidden states at step t; for AGNN, we use $\boldsymbol{H}^t(x)$ of size $|V_{G(x)}| \times D$ to denote.

[1]Embedding-based methods generally perform better than path-based methods.

We define the objective based on our two GNNs as $\sum_{i=1}^{N} l(x_i, y_i; \boldsymbol{\theta}_{G(x_i)}, \boldsymbol{\theta}_{\mathcal{G}})$, where $G(x_i)$ is dynamically constructed. First, we write the standard message passing in IGNN as

$$\boldsymbol{\mathcal{H}}^t = f_{\text{IGNN}}(\boldsymbol{\mathcal{H}}^{t-1}; \boldsymbol{\theta}_{\mathcal{G}}), \tag{1}$$

where f_{IGNN} represents all involved operations in one message passing iteration over $\mathcal{G}$, including: (1) computing messages along each edge with the complexity[2] of $\mathcal{O}(BD|\mathcal{E}|)$, (2) aggregating messages received at each node with the complexity of $\mathcal{O}(BD|\mathcal{E}|)$, and (3) updating node hidden states with the complexity of $\mathcal{O}(BD|\mathcal{V}|)$. For a T-step propagation, we get the per-batch complexity of $\mathcal{O}(BDT(|\mathcal{E}| + |\mathcal{V}|))$. Considering that backpropagation requires intermediate computation results to be saved during one pass, this complexity counts for both time and space. However, since IGNN is input-invariant, its node representations can be shared across input examples in one batch so that B can be removed to get $\mathcal{O}(DT(|\mathcal{E}| + |\mathcal{V}|))$. If we sample a smaller set $\hat{\mathcal{E}}$ from $\mathcal{E}$ to run such that $|\hat{\mathcal{E}}| \approx k|\mathcal{V}|$, we can further reduce the complexity to $\mathcal{O}(DT|\mathcal{V}|)$.

The pruned message passing in AGNN can be written as

$$\boldsymbol{H}^t(x) = f_{\text{AGNN}}(\boldsymbol{H}^{t-1}(x), \boldsymbol{\mathcal{H}}^t; \boldsymbol{\theta}_{G(x)}). \tag{2}$$

Its complexity can be computed similarly as above. However, we cannot remove B. Fortunately, subgraph $G(x)$ is not $\mathcal{G}$. If we let x be a node v, $G(x)$ grows from a single node, i.e., $G^0(x) = \{v\}$, and expands itself each step, leading to a sequence of $(G^0(x), G^1(x), \ldots, G^T(x))$. Here, we describe the expansion behavior as *consecutive expansion*, which means no jumping across neighborhood allowed, so that we can ensure that

$$G^t(x) \subseteq G^{t-1}(x) \cup \partial G^{t-1}(x) \subseteq G^{t-2}(x) \cup \partial^2 G^{t-2}(x). \tag{3}$$

Many real-world graphs follow the *small-world* pattern, and the *six degrees of separation* implies $G^0(x) \cup \partial^6 G^0(x) \approx \mathcal{G}$. The upper bound of $G^t(x)$ can grow exponentially in t, and there is no guarantee that $G^t(x)$ will not explode.

Proposition. *Given a graph $\mathcal{G}$ (undirected or directed in both directions), we assume the probability of the degree of an arbitrary node being less than or equal to d is larger than p, i.e., $P(\deg(v) \leq d) > p, \forall v \in V$. Considering a sequence of consecutively expanding subgraphs $(G^0, G^1, \ldots, G^T)$, starting with $G^0 = \{v\}$, for all $t \geq 1$, we can ensure*

$$P\big(|V_{G^t}| \leq \frac{d(d-1)^t - 2}{d-2}\big) > p^{\frac{d(d-1)^{t-1}-2}{d-2}}. \tag{4}$$

The proposition implies the guarantee of upper-bounding $|V_{G^t(x)}|$ becomes exponentially looser and weaker as t gets larger even if the given assumption has a small d and a large p (close to 1). We define graph increment at step t as $\Delta G^t(x)$ such that $G^t(x) = G^{t-1}(x) \cup \Delta G^t(x)$. To prevent $G^t(x)$ from explosion, we need to constrain $\Delta G^t(x)$. We propose several sampling strategies:

1. $\Delta G^t(x) = \hat{\partial} G^{t-1}(x)$, which means we sample nodes from the boundary of $G^{t-1}(x)$.
2. $\Delta G^t(x) = \partial \widehat{G^{t-1}(x)}$, which means we take the boundary of sampled nodes from $G^{t-1}(x)$.
3. $\Delta G^t(x) = \hat{\partial} \widehat{G^{t-1}(x)}$, which means we sample nodes from the boundary of $\widehat{G^{t-1}(x)}$.
4. $\Delta G^t(x) = \widehat{\hat{\partial} \widehat{G^{t-1}(x)}}$, which means we samples nodes from $\hat{\partial} \widehat{G^{t-1}(x)}$.

Obviously, we have $\widehat{\hat{\partial} \widehat{G^{t-1}(x)}} \subseteq \hat{\partial} \widehat{G^{t-1}(x)} \subseteq \partial \widehat{G^{t-1}(x)}$ and $G^{t-1}(x) \cup \partial \widehat{G^{t-1}(x)} \subseteq G^{t-1}(x) \cup \partial G^{t-1}(x)$. Further, we let N_g and N_s be the maximum number of sampled nodes in $\partial \widehat{G^{t-1}(x)}$ and the last sampling of $\widehat{\hat{\partial} \widehat{G^{t-1}(x)}}$ respectively and let N_b be per-node maximum sampled neighbors in $\hat{\partial} G^{t-1}(x)$, and then we can obtain much tighter guarantee as follow:

1. $P(|V_{\Delta G^t(x)}| \leq N_g(d-1)) > p^{N_g}$ for $\partial \widehat{G^{t-1}(x)}$.
2. $P(|V_{\Delta G^t(x)}| \leq N_g N_b) = 1$ and $P(|V_{\Delta G^t(x)}| \leq N_g \cdot \min(d-1, N_b)) > p^{N_g}$ for $\hat{\partial} \widehat{G^{t-1}(x)}$.

[2] We assume per-example per-edge per-dimension time cost as a unit time.

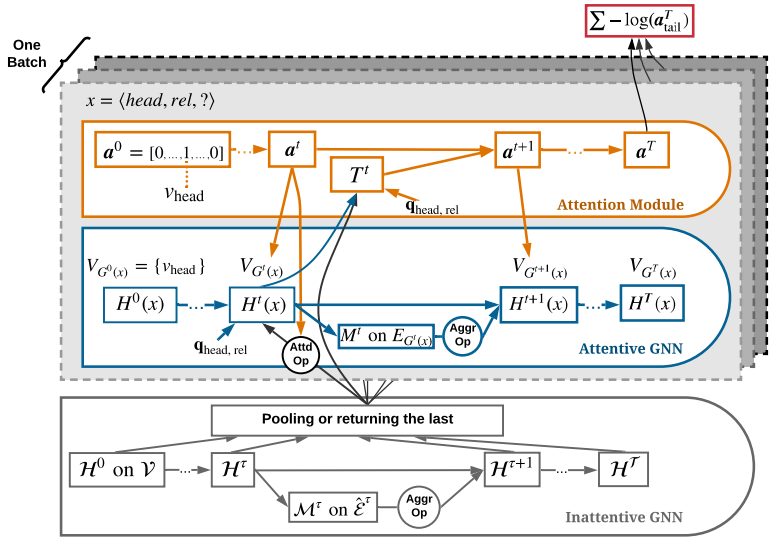


Figure 2: Model architecture used in knowledge graph reasoning.

3. $P(|V_{\Delta G^t(x)}| \leq \min(N_g N_b, N_s)) = 1$ for $\widehat{\hat{\partial} G^{t-1}(x)}$.

By $\widehat{\hat{\partial} G^{t-1}(x)}$, we can guarantee $|V_{G^T(x)}| \leq 1 + T \min(N_g N_b, N_s)$. To constrain the growth of $G^{t-1}(x)$, we can decrease either $N_g N_b$ or N_s. However, smaller sample size means less area explored and less chance to hit target nodes. We thus use attention operations to do the top-K selection instead of random sampling when K has to be small. We change $\widehat{\hat{\partial} G^{t-1}(x)}$ to $\widetilde{\hat{\partial} G^{t-1}(x)}$ where $\sim$ represents the operation of attending over nodes and picking the top-K. There are two types of attention operations, one applied to $G^{t-1}(x)$ and the other applied to $\widetilde{\hat{\partial} G^{t-1}(x)}$. Note that the size of $\widetilde{\hat{\partial} G^{t-1}(x)}$ might be much larger than $\widetilde{G^{t-1}(x)}$ if we want to sample more nodes with larger N_b to sufficiently explore the boundary, $\widetilde{\partial G^{t-1}(x)}$. Nevertheless, we can address this problem by using small dimensions to compute attention scores, since attention carried by each node is just a scalar, much smaller than a node representation vector computed during message passing over $G^{t-1}(x)$.

3 Model Implementation

3.1 Architecture design for knowledge graph reasoning

Our model architecture as shown in Figure 2 consists of:

- *IGNN module*: performs standard message passing to compute full-graph node representations.
- *AGNN module*: performs a batch of pruned message passing to compute input-dependent node representations which also make use of low-level representations from IGNN.
- *Attention Module*: performs a flow-style attention transition process, conditioned on node representations from both IGNN and AGNN but only affecting AGNN.

We let $\langle \mathcal{V}, \mathcal{E} \rangle$ denote a knowledge graph where $\mathcal{V}$ is a set of entities and $\mathcal{E}$ is a set of relations. Each edge or relation is represented by a triple $\langle head, rel, tail \rangle$, where $head$ is the head entity, $tail$ is the tail entity, and rel is their relation type. The goal is to predict potential unknown links, i.e., which entity is likely to be the tail given a query $\langle head, rel, ? \rangle$ with the head and the relation type specified.

IGNN module. We implement it using standard message passing mechanism [21]. If the full graph has an extremely large number of edges, we sample a subset of edges, $\hat{\mathcal{E}}^\tau \subset \mathcal{E}$, randomly each step. For a batch of input queries, we let node representations from IGNN be shared across queries, containing no batch dimension. Thus, its complexity does not scale with batch size and the saved computation resources can be allocated to sampling more edges. Each node v has a state $\boldsymbol{\mathcal{H}}^\tau_{v,:}$ at step

τ, where the initial $\boldsymbol{\mathcal{H}}^0_{v,:} = \boldsymbol{e}_v$. Each edge $\langle v', r, v\rangle$ produces a message, denoted by $\boldsymbol{\mathcal{M}}^\tau_{\langle v',r,v\rangle,:}$ at step τ. The computation components include:

- Message function: $\boldsymbol{\mathcal{M}}^\tau_{\langle v',r,v\rangle,:} = \psi_{\text{IGNN}}(\boldsymbol{\mathcal{H}}^\tau_{v',:}, \boldsymbol{e}_r, \boldsymbol{\mathcal{H}}^\tau_{v,:})$, where $\langle v', r, v\rangle \in \hat{\mathcal{E}}^\tau$.
- Message aggregation: $\overline{\boldsymbol{\mathcal{M}}}^\tau_{v,:} = \frac{1}{\sqrt{N^\tau(v)}} \sum_{v',r} \boldsymbol{\mathcal{M}}^\tau_{\langle v',r,v\rangle,:}$, where $\langle v', r, v\rangle \in \hat{\mathcal{E}}^\tau$.
- Node state update function: $\boldsymbol{\mathcal{H}}^{\tau+1}_{v,:} = \boldsymbol{\mathcal{H}}^\tau_{v,:} + \delta_{\text{IGNN}}(\boldsymbol{\mathcal{H}}^\tau_{v,:}, \overline{\boldsymbol{\mathcal{M}}}^\tau_{v,:}, \boldsymbol{e}_v)$, where $v \in \mathcal{V}$.

We compute messages only for the sampled edges, $\langle v', r, v\rangle \in \hat{\mathcal{E}}^\tau$, each step. Functions ψ_{IGNN} and δ_{IGNN} are implemented by a two-layer MLP (using leakyReLu for the first layer and $\tanh$ for the second) with input arguments concatenated respectively. Messages are aggregated by dividing the sum by the square root of $N^\tau(v)$, the number of sampled neighbors that send messages to v, preserving the scale of variance. We use a residual adding to update each node state instead of a GRU or a LSTM. After running for $\mathcal{T}$ steps, we output a pooling result or simply the last, denoted by $\boldsymbol{\mathcal{H}} = \boldsymbol{\mathcal{H}}^{\mathcal{T}}$, to feed into downstream modules.

AGNN module. AGNN is input-dependent, which means node states depend on input query $x = \langle head, rel, ?\rangle$, denoted by $\boldsymbol{H}^t_{v,:}(x)$. We implement pruned message passing, running on small subgraphs each conditioned on an input query. We leverage the sparsity and only save $\boldsymbol{H}^t_{v,:}(x)$ for visited nodes $v \in V_{G^t(x)}$. When $t = 0$, we start from node $head$ with $V_{G^0(x)} = \{v_{head}\}$. When computing messages, denoted by $\boldsymbol{M}^t_{\langle v',r,v\rangle,:}(x)$, we use a sampling-attending procedure, explained in Section 3.2, to constrain the number of computed edges. The computation components include:

- Message function: $\boldsymbol{M}^t_{\langle v',r,v\rangle,:}(x) = \psi_{\text{AGNN}}(\boldsymbol{H}^t_{v',:}(x), \boldsymbol{c}_r(x), \boldsymbol{H}^t_{v,:}(x))$, where $\langle v', r, v\rangle \in E_{G^t(x)}$[3], and $\boldsymbol{c}_r(x) = [\boldsymbol{e}_r, \boldsymbol{q}_{head}, \boldsymbol{q}_{rel}]$ represents a context vector.
- Message aggregation: $\overline{\boldsymbol{M}}^t_{v,:}(x) = \frac{1}{\sqrt{N^t(v)}} \sum_{v',r} \boldsymbol{M}^t_{\langle v',r,v\rangle,:}(x)$, where $\langle v', r, v\rangle \in E_{G^t(x)}$.
- Node state attending function: $\widetilde{\boldsymbol{H}}^{t+1}_{v,:}(x) = a^{t+1}_v \boldsymbol{W} \boldsymbol{\mathcal{H}}_{v,:}$, where a^{t+1}_v is an attention score.
- Node state update function: $\boldsymbol{H}^{t+1}_{v,:}(x) = \boldsymbol{H}^t_{v,:}(x) + \delta_{\text{AGNN}}(\boldsymbol{H}^t_{v,:}(x), \overline{\boldsymbol{M}}^t_{v,:}(x), \boldsymbol{c}^{t+1}_v(x))$, where $\boldsymbol{c}^{t+1}_v(x) = [\widetilde{\boldsymbol{H}}^{t+1}_{v,:}(x), \boldsymbol{q}_{head}, \boldsymbol{q}_{rel}]$ also represents a context vector.

Query context is defined by its head and relation type embeddings, i.e., $\boldsymbol{q}_{head} = \boldsymbol{e}_{head}$ and $\boldsymbol{q}_{rel} = \boldsymbol{e}_{rel}$. We introduce a node state attending function to pass node representation information from IGNN to AGNN weighted by a scalar attention score a^{t+1}_v and projected by a learnable matrix $\boldsymbol{W}$. We initialize $\boldsymbol{H}^0_{v,:}(x) = \boldsymbol{\mathcal{H}}_{v,:}$ for node $v \in V_{G^0(x)}$, treating the rest as zero states.

Attention module. Attention over T steps is represented by a sequence of node probability distributions, denoted by $\boldsymbol{a}^t$ $(t = 1, 2 \ldots, T)$. The initial distribution $\boldsymbol{a}^0$ is a one-hot vector with $\boldsymbol{a}^0[v_{head}] = 1$. To spread attention, we need to compute transition matrices $\boldsymbol{T}^t$ each step. Since it is conditioned on both IGNN and AGNN, we capture two types of interaction between v' and v: $\boldsymbol{H}^t_{v',:}(x) \sim \boldsymbol{H}^t_{v,:}(x)$, and $\boldsymbol{H}^t_{v',:}(x) \sim \boldsymbol{\mathcal{H}}_{v,:}$. The former favors visited nodes, while the latter is used to attend to unseen nodes.

$$
\begin{aligned}
\boldsymbol{T}^t_{:,v'} &= \text{softmax}_{v \in N^t(v')}\big(\sum\nolimits_r \alpha_1(\boldsymbol{H}^t_{v',:}(x), \boldsymbol{c}_r(x), \boldsymbol{H}^t_{v,:}(x)) + \alpha_2(\boldsymbol{H}^t_{v',:}(x), \boldsymbol{c}_r(x), \boldsymbol{\mathcal{H}}_{v,:})\big) \\
\alpha_1(\cdot) &= \text{MLP}(\boldsymbol{H}^t_{v',:}(x), \boldsymbol{c}_r(x))^{\text{T}} \boldsymbol{W}_1 \text{MLP}(\boldsymbol{H}^t_{v,:}(x), \boldsymbol{c}_r(x)) \\
\alpha_2(\cdot) &= \text{MLP}(\boldsymbol{H}^t_{v',:}(x), \boldsymbol{c}_r(x))^{\text{T}} \boldsymbol{W}_2 \text{MLP}(\boldsymbol{\mathcal{H}}_{v,:}, \boldsymbol{c}_r(x))
\end{aligned}
\tag{5}
$$

where $\boldsymbol{W}_1$ and $\boldsymbol{W}_2$ are two learnable matrices. Each MLP uses one single layer with the leakyReLu activation. To reduce the complexity for computing $\boldsymbol{T}^t$, we use nodes $v' \in \widetilde{V_{G^t(x)}}$, which contains nodes with the k-largest attention scores at step t, and use nodes v sampled from v''s neighbors to compute attention transition for the next step. Due to the fact that nodes v' result from the top-k pruning, the loss of attention may occur to diminish the total amount. Therefore, we use a renormalized version, $\boldsymbol{a}^{t+1} = \boldsymbol{T}^t \boldsymbol{a}^t / \|\boldsymbol{T}^t \boldsymbol{a}^t\|$, to compute new attention scores. We use attention scores at the final step as the probability to predict the tail node.

[3] In practice, we can use a smaller set of edges than $E_{G^t(x)}$ to pass messages as discussed in Section 3.2

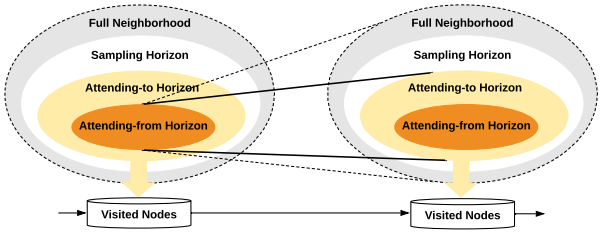


Figure 3: Iterative sampling-attending procedure balancing between coverage and complexity.

3.2 Complexity reduction by iterative sampling and attending

AGNN deals with local subgraphs for each input x so that only a few selected nodes are kept in $V_{G^t(x)}$, called *visited nodes*, and $|V_{G^t(x)}|$ is much smaller than $|\mathcal{V}|$. The initial $V_{G^0(x)}$ contains only one node v_{head}, and then $V_{G^t(x)}$ is enlarged each step by adding new nodes. When propagating messages, we can just consider the one-step neighborhood each step. However, the expansion goes so rapidly that it covers almost all nodes after a few steps. The key to address the problem is to constrain the scope of nodes we can expand the boundary from, i.e., the core nodes which determine where we can go next. We call it the *attending-from horizon*, $\widetilde{G^t(x)}$, selected according to attention scores $\boldsymbol{a}^t$. Given this horizon, we still need edge sampling over its neighborhood instead of using the whole $N(\widetilde{G^t(x)})$ in case of a hub node of extremely high degree. Here, we face a trade-off between coverage and complexity when sampling over the neighborhood. Also, we need node representations within each subgraph to keep their information coherent and avoid possible noises caused by randomly sampling. Therefore, we introduce an *attending-to horizon* inside the *sampling horizon*. We denote the sampling horizon by $\widehat{N}(\widetilde{G^t(x)})$ and the attending-to horizon by $\widetilde{\widehat{N}(\widetilde{G^t(x)})}$. The attention module runs within the sampling horizon with smaller dimensions in order to sample more neighbors for a larger coverage. Then, we prune the sampling horizon to obtain the attending-to horizon, which contains a subset of nodes selected according to newly computed attention scores $\boldsymbol{a}^{t+1}$. Current message passing iteration at step t in AGNN can be further constrained on edges between $\widetilde{G^t(x)}$ and $\widetilde{\widehat{N}(\widetilde{G^t(x)})}$, a smaller set than $E_{G^t(x)}$. We illustrate this procedure in Figure 3.

4 Experiments

Datasets. We use six large KG datasets: FB15K, FB15K-237, WN18, WN18RR, NELL995, and YAGO3-10. FB15K-237 [28] is sampled from FB15K [6] with redundant relations removed, and WN18RR [8] is a subset of WN18 [6] removing triples that cause test leakage. Thus, they are both considered more challenging. NELL995 [13] has separate datasets for 12 query relations each corresponding to a single-query-relation KBC task. YAGO3-10 [29] contains the largest KG with millions of edges. Their statistics are shown in Table 1. We find some statistical differences between train and validation (or test). In a KG with all training triples as its edges, a triple $(head, rel, tail)$ is considered as a multi-edge triple if the KG contains other triples that also connect $head$ and $tail$ ignoring the direction. We notice that FB15K-237 is a special case compared to the others, as there are no edges in its KG directly linking any pair of $head$ and $tail$ in validation (or test). Therefore, when using training triples as queries to train our model, given a batch, for FB15K-237, we cut off from the KG all triples connecting the head-tail pairs in the given batch, ignoring relation types and edge directions, forcing the model to learn a composite reasoning pattern rather than a single-hop pattern, and for the rest datasets, we only remove the triples of this batch and their inverse from the KG to avoid information leakage before training on this batch. This can be regarded as a hyperparameter tuning whether to force a multi-hop reasoning or not, leading to a performance boost of about 2% in HITS@1 on FB15-237.

Experimental settings. We use the same data split protocol as in many papers [8, 13, 14]. We create a KG, a directed graph, consisting of all train triples and their inverse added for each dataset except NELL995, since it already includes reciprocal relations. Besides, every node in KGs has a self-loop edge to itself. We also add inverse relations into the validation and test set to evaluate the two directions. For evaluation metrics, we use HITS@1,3,10 and the mean reciprocal rank (MRR) in the filtered setting for FB15K-237, WN18RR, FB15K, WN18, and YAGO3-10, and use the mean average precision (MAP) for NELL995's single-query-relation KBC tasks. For NELL995, we follow

Table 1: Statistics of the six KG datasets. PME (tr) means the proportion of multi-edge triples in train; PME (va) means the proportion of multi-edge triples in validation; AL (va) means the average length of shortest paths connecting each head-tail pair in validation.

Dataset	#Entities	#Rels	#Train	#Valid	#Test	PME (tr)	PME (va)	AL (va)
FB15K	14,951	1,345	483,142	50,000	59,071	81.2%	80.6%	1.22
FB15K-237	14,541	237	272,115	17,535	20,466	38.0%	**0%**	2.25
WN18	40,943	18	141,442	5,000	5,000	93.1%	94.0%	1.18
WN18RR	40,943	11	86,835	3,034	3,134	34.5%	35.5%	2.84
NELL995	74,536	200	149,678	543	2,818	100%	31.1%	2.00
YAGO3-10	123,188	37	1,079,040	5,000	5,000	56.4%	56.0%	1.75

Table 2: Comparison results on the FB15K-237 and WN18RR datasets. Results of [♠] are taken from [30], [♣] from [8], [♡] from [15], [♢] from [10], [△] from [14], and [✠] from [11]. Some collected results only have a metric score while some including ours take the form of "mean (std)".

	FB15K-237				**WN18RR**			
Metric (%)	H@1	H@3	H@10	MRR	H@1	H@3	H@10	MRR
TransE [♠]	-	-	46.5	29.4	-	-	50.1	22.6
DistMult [♣]	15.5	26.3	41.9	24.1	39	44	49	43
DistMult [♡]	20.6 (.4)	31.8 (.2)	-	29.0 (.2)	38.4 (.4)	42.4 (.3)	-	41.3 (.3)
ComplEx [♣]	15.8	27.5	42.8	24.7	41	46	51	44
ComplEx [♡]	20.8 (.2)	32.6 (.5)	-	29.6 (.2)	38.5 (.3)	43.9 (.3)	-	42.2 (.2)
ConvE [♣]	23.7	35.6	50.1	32.5	40	44	52	43
ConvE [♡]	23.3 (.4)	33.8 (.3)	-	30.8 (.2)	39.6 (.3)	44.7 (.2)	-	43.3 (.2)
RotatE [♢]	24.1	37.5	53.3	33.8	42.8	49.2	**57.1**	47.6
ComplEx-N3[✠]	-	-	**56**	**37**	-	-	57	48
NeuralLP [♡]	18.2 (.6)	27.2 (.3)	-	24.9 (.2)	37.2 (.1)	43.4 (.1)	-	43.5 (.1)
MINERVA [♡]	14.1 (.2)	23.2 (.4)	-	20.5 (.3)	35.1 (.1)	44.5 (.4)	-	40.9 (.1)
MINERVA [△]	-	-	45.6	-	41.3	45.6	51.3	-
M-Walk [♡]	16.5 (.3)	24.3 (.2)	-	23.2 (.2)	41.4 (.1)	44.5 (.2)	-	43.7 (.1)
DPMPN	**28.6 (.1)**	**40.3 (.1)**	53.0 (.3)	36.9 (.1)	**44.4 (.4)**	**49.7 (.8)**	55.8 (.5)	**48.2 (.5)**

the same evaluation procedure as in [13, 14, 15], ranking the answer entities against the negative examples given in their experiments. We run our experiments using a 12G-memory GPU, TITAN X (Pascal), with Intel(R) Xeon(R) CPU E5-2670 v3 @ 2.30GHz. Our code is written in Python based on TensorFlow 2.0 and NumPy 1.16 and can be found by the link[4] below. We run three times for each hyperparameter setting per dataset to report the means and standard deviations. See hyperparameter details in the appendix.

Baselines. We compare our model against embedding-based approaches, including TransE [6], TransR [31], DistMult [7], ConvE [8], ComplE [9], HolE [32], RotatE [10], and ComplEx-N3 [11], and path-based approaches that use RL methods, including DeepPath [13], MINERVA [14], and M-Walk [15], and also that uses learned neural logic, NeuralLP [19].

Comparison results and analysis. We report comparison on FB15K-23 and WN18RR in Table 2. Our model DPMPN significantly outperforms all the baselines in HITS@1,3 and MRR. Compared to the best baseline, we only lose a few points in HITS@10 but gain a lot in HITS@1,3. We speculate that it is the reasoning capability that helps DPMPN make a sharp prediction by exploiting graph-structured composition locally and conditionally. When a target becomes too vague to predict, reasoning may lose its advantage against embedding-based models. However, path-based baselines, with a certain ability to do reasoning, perform worse than we expect. We argue that it might be inappropriate to think of reasoning, a sequential decision process, equivalent to a sequence of nodes. The average lengths of the shortest paths between heads and tails as shown in Table 1 suggests a very short path, which makes the motivation of using a path almost useless. The reasoning pattern should be modeled in the form of dynamical local graph-structured pattern with nodes densely connected with each other to produce a decision collectively. We also run our model on FB15K, WN18, and YAGO3-10, and the comparison results in the appendix show that DPMPN achieves a very competitive position against the best state of the art. We summarize the comparison on NELL995's tasks in the appendix. DPMPN performs the best on five tasks, also being competitive on the rest.

[4]https://github.com/netpaladinx/DPMPN

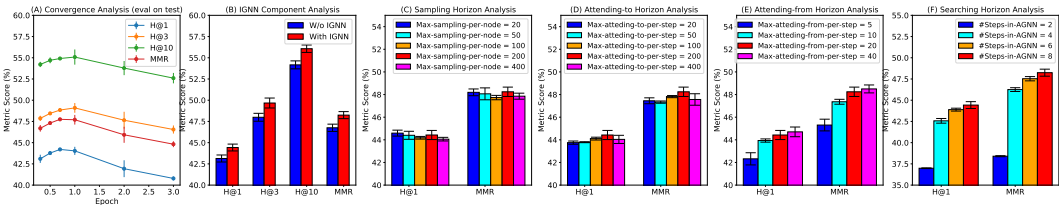


Figure 4: Experimental analysis on WN18RR. (A) Convergence analysis: we pick six model snapshots during training and evaluate them on test. (B) IGNN component analysis: *w/o IGNN* uses zero step to run message passing, while *with IGNN* uses two; (C)-(F) Sampling, attending-to, attending-from and searching horizon analysis. The charts on FB15K-237 can be found in the appendix.

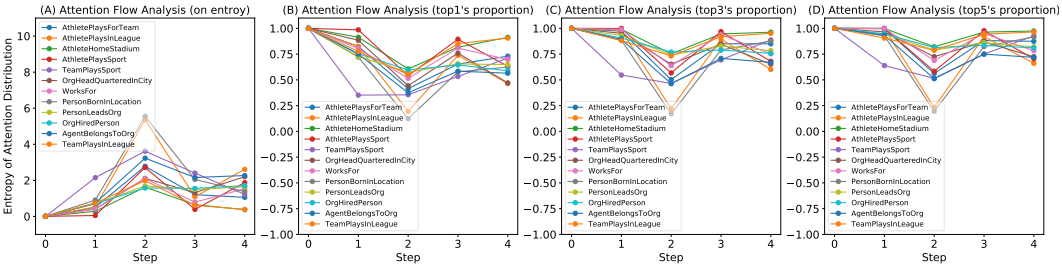


Figure 5: Analysis of attention flow on NELL995 tasks. (A) The average entropy of attention distributions changing along steps for each single-query-relation KBC task. (B)(C)(D) The changing of the proportion of attention concentrated at the top-1,3,5 nodes per step for each task.

Convergence analysis. Our model converges very fast during training. We may use half of training queries to train model to generalize as shown in Figure 4(A). Compared to less expensive embedding-based models, our model need to traverse a number of edges when training on one input, consuming much time per batch, but it does not need to pass a second epoch, thus saving a lot of training time. The reason may be that training queries also belong to the KG's edges and some might be exploited to construct subgraphs during training on other queries.

Module analysis. If we do not run message passing in IGNN, $\boldsymbol{\mathcal{H}}_{v,:}$ is just the initial embedding of node v, and we can still run pruned message passing in AGNN as usual. We want to know whether IGNN is actually useful. Considering that long-range propagated messages might bring in noisy features, we compare running IGNN for two steps against totally shutting it down. The result in Figure 4(B) shows that IGNN brings a small gain in each metric on WN18RR.

Horizon analysis. The sampling, attending-to, attending-from and searching (i.e., propagation steps) horizons determine how large area a subgraph can expand over. These factors affect computation complexity as well as prediction performance. Intuitively, enlarging the exploring area by sampling more, attending more, and searching longer, may increase the chance of hitting a target to gain some performance. However, the experimental results in Figure 4(C)(D) show that it is not always the case. In Figure 4(E), we can see that increasing the maximum number of attending-from nodes per step is useful, but normal GPUs with a limited memory do not allow for an arbitrarily large number due to heavy intermediate data produced during feedforward computing. Figure 4(F) suggests that the propagation steps of AGNN should not go below four.

Attention flow analysis. If the flow-style attention really captures the way we reason about the world, its process should be conducted in a diverging-converging thinking pattern. Intuitively, first, for the diverging thinking phase, we search and collect ideas as much as we can; then, for the converging thinking phase, we try to concentrate our thoughts on one point. To check whether the attention flow has such a pattern, we measure the average entropy of attention distributions changing along steps and also the proportion of attention concentrated at the top-1,3,5 nodes. As we expect, attention is more focused at the final step and the beginning.

Time cost analysis. The time cost is affected not only by the scale of a dataset but also by the horizon setting. For each dataset, we list the training time for one epoch corresponding to our standard hyperparameter settings in the appendix. Note that there is always a trade-off between complexity and performance. We thus study whether we can reduce time cost a lot at the price of sacrificing a little performance. We plot the one-epoch training time in Figure 6(A)-(D), using the same settings as

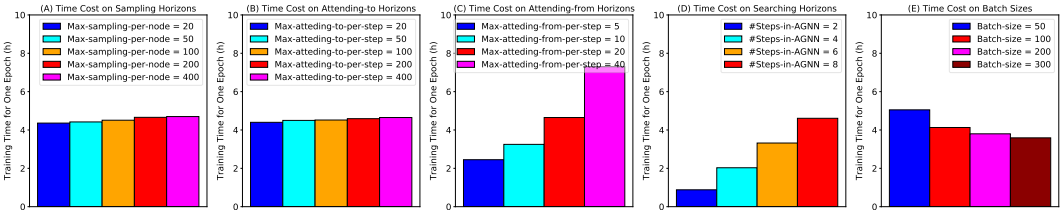


Figure 6: Analysis of time cost on WN18RR: (A)-(D) measure the one-epoch training time on different horizon settings corresponding to Figure 4(C)-(F); (E) measures on different batch sizes using horizon setting *Max-sampling-per-node*=20, *Max-attending-to-per-step*=20, *Max-attending-from-per-step*=20, and *#Steps-in-AGNN*=8. The charts on FB15K-237 can be found in the appendix.

we do in the horizon analysis. We can see that *Max-attending-from-per-step* and *#Steps-in-AGNN* affect the training time significantly while *Max-sampling-per-node* and *Max-attending-to-per-step* affect very slightly. Therefore, we can use smaller *Max-sampling-per-node* and *Max-attending-to-per-step* in order to gain a larger batch size, making the computation more efficiency as shown in Figure 6(E).

Visualization. To further demonstrate the reasoning capability, we show visualization results of some pruned subgraphs on NELL995's test data for 12 separate tasks. We avoid using the training data in order to show generalization of the learned reasoning capability. We show the visualization results in Figure 1. See the appendix for detailed analysis and more visualization results.

5 Related Work

Knowledge graph reasoning. Early work, including TransE [6] and its analogues [33, 31, 34], DistMult [7], ConvE [8] and ComplEx [9], focuses on learning embeddings of entities and relations. Some recent works of this line [10, 11] achieve high accuracy. Another line aims to learn inference paths [35, 36, 37, 38, 39, 40] for knowledge graph reasoning, especially DeepPath [13], MINERVA [14], and M-Walk [15], which use RL to learn multi-hop relational paths. However, these approaches, based on policy gradients or Monte Carlo tree search, often suffer from low sample efficiency and sparse rewards, requiring a large number of rollouts and sophisticated reward function design. Other efforts include learning soft logical rules [18, 19] or compostional programs [41].

Relational reasoning in Graph Neural Networks. Relational reasoning is regarded as the key for combinatorial generalization, taking the form of entity- and relation-centric organization to reason about the composition structure of the world [42, 43]. A multitude of recent implementations [4] encode relational inductive biases into neural networks to exploit graph-structured representation, including graph convolution networks (GCNs) [44, 45, 46, 47, 48, 49, 50, 51] and graph neural networks [20, 52, 53, 54, 21]. Variants of GNN architectures have been developed. Relation networks [53] use a simple but effective neural module to model relational reasoning, and its recurrent versions [55, 56] do multi-step relational inference for long periods; Interaction networks [54] provide a general-purpose learnable physics engine, and two of its variants are visual interaction networks [57] and vertex attention interaction networks [24]; Message passing neural networks [21] unify various GCNs and GNNs into a general message passing formalism by analogy to the one in graphical models.

Attention mechanism on graphs. Neighborhood attention operation can enhance GNNs' representation power [23, 24, 58, 59]. These approaches often use multi-head self-attention to focus on specific interactions with neighbors when aggregating messages, inspired by [60, 61, 62]. Most graph-based attention mechanisms attend over neighborhood in a single-hop fashion, and [24] claims that the multi-hop architecture does not help to model high-order interaction in experiments. However, a flow-style design of attention in [63] shows a way to model long-range attention, stringing isolated attention operations by transition matrices.

6 Conclusion

We introduce *Dynamically Pruned Message Passing Networks* (DPMPN) and apply it to large-scale knowledge graph reasoning tasks. We propose to learn an input-dependent local subgraph which is progressively and selectively constructed to model a sequential reasoning process in knowledge graphs. We use graphical attention expression, a flow-style attention mechanism, to guide and prune

the underlying message passing, making it scalable for large-scale graphs and also providing clear graphical interpretations. We also take the inspiration from the consciousness prior to develop a two-GNN framework to boost experimental performances.

References

[1] Yoshua Bengio. The consciousness prior. *CoRR*, abs/1709.08568, 2017.

[2] Judea Pearl and Dana Mackenzie. *The book of why: the new science of cause and effect*. Basic Books, 2018.

[3] Yoshua Bengio. Challenges for deep learning towards human-level ai, 11 2018.

[4] Peter W. Battaglia, Jessica B. Hamrick, Victor Bapst, Alvaro Sanchez-Gonzalez, Vinícius Flores Zambaldi, Mateusz Malinowski, Andrea Tacchetti, David Raposo, Adam Santoro, Ryan Faulkner, Çaglar Gülçehre, Francis Song, Andrew J. Ballard, Justin Gilmer, George E. Dahl, Ashish Vaswani, Kelsey R. Allen, Charles Nash, Victoria Langston, Chris Dyer, Nicolas Heess, Daan Wierstra, Pushmeet Kohli, Matthew Botvinick, Oriol Vinyals, Yujia Li, and Razvan Pascanu. Relational inductive biases, deep learning, and graph networks. *CoRR*, abs/1806.01261, 2018.

[5] Keyulu Xu, Jingling Li, Mozhi Zhang, Simon S. Du, Ken ichi Kawarabayashi, and Stefanie Jegelka. What can neural networks reason about? *ArXiv*, abs/1905.13211, 2019.

[6] Antoine Bordes, Nicolas Usunier, Alberto García-Durán, Jason Weston, and Oksana Yakhnenko. Translating embeddings for modeling multi-relational data. In *NIPS*, 2013.

[7] Bishan Yang, Wen tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. Embedding entities and relations for learning and inference in knowledge bases. *CoRR*, abs/1412.6575, 2015.

[8] Tim Dettmers, Pasquale Minervini, Pontus Stenetorp, and Sebastian Riedel. Convolutional 2d knowledge graph embeddings. In *AAAI*, 2018.

[9] Théo Trouillon, Johannes Welbl, Sebastian Riedel, Éric Gaussier, and Guillaume Bouchard. Complex embeddings for simple link prediction. In *ICML*, 2016.

[10] Zhiqing Sun, Zhi-Hong Deng, Jian-Yun Nie, and Jian Tang. Rotate: Knowledge graph embedding by relational rotation in complex space. *CoRR*, abs/1902.10197, 2018.

[11] Timothée Lacroix, Nicolas Usunier, and Guillaume Obozinski. Canonical tensor decomposition for knowledge base completion. In *ICML*, 2018.

[12] William Wang. Knowledge graph reasoning: Recent advances, 2018.

[13] Wenhan Xiong, Thien Hoang, and William Yang Wang. Deeppath: A reinforcement learning method for knowledge graph reasoning. In *EMNLP*, 2017.

[14] Rajarshi Das, Shehzaad Dhuliawala, Manzil Zaheer, Luke Vilnis, Ishan Durugkar, Akshay Krishnamurthy, Alexander J. Smola, and Andrew McCallum. Go for a walk and arrive at the answer: Reasoning over paths in knowledge bases using reinforcement learning. *CoRR*, abs/1711.05851, 2018.

[15] Yelong Shen, Jianshu Chen, Pu Huang, Yuqing Guo, and Jianfeng Gao. M-walk: Learning to walk over graphs using monte carlo tree search. In *NeurIPS*, 2018.

[16] Wenhu Chen, Wenhan Xiong, Xifeng Yan, and William Yang Wang. Variational knowledge graph reasoning. In *NAACL-HLT*, 2018.

[17] Xi Victoria Lin, Richard Socher, and Caiming Xiong. Multi-hop knowledge graph reasoning with reward shaping. In *EMNLP*, 2018.

[18] William W. Cohen. Tensorlog: A differentiable deductive database. *CoRR*, abs/1605.06523, 2016.

[19] Fan Yang, Zhilin Yang, and William W. Cohen. Differentiable learning of logical rules for knowledge base reasoning. In *NIPS*, 2017.

[20] Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. The graph neural network model. *IEEE Transactions on Neural Networks*, 20:61–80, 2009.

[21] Justin Gilmer, Samuel S. Schoenholz, Patrick F. Riley, Oriol Vinyals, and George E. Dahl. Neural message passing for quantum chemistry. In *ICML*, 2017.

[22] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? *ArXiv*, abs/1810.00826, 2018.

[23] Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Alejandro Romero, Pietro Lió, and Yoshua Bengio. Graph attention networks. *CoRR*, abs/1710.10903, 2018.

[24] Yedid Hoshen. Vain: Attentional multi-agent predictive modeling. In *NIPS*, 2017.

[25] Danilo Jimenez Rezende and Shakir Mohamed. Variational inference with normalizing flows. *ArXiv*, abs/1505.05770, 2015.

[26] Stanislas Dehaene, Michel Kerszberg, and Jean Pierre Changeux. A neuronal model of a global workspace in effortful cognitive tasks. *Proceedings of the National Academy of Sciences of the United States of America*, 95 24:14529–34, 1998.

[27] Giulio Tononi, Mélanie Boly, Marcello Massimini, and Christof Koch. Integrated information theory: from consciousness to its physical substrate. *Nature Reviews Neuroscience*, 17:450–461, 2016.

[28] Kristina Toutanova and Danqi Chen. Observed versus latent features for knowledge base and text inference. In *Proceedings of the 3rd Workshop on Continuous Vector Space Models and their Compositionality*, 2015.

[29] Farzaneh Mahdisoltani, Joanna Asia Biega, and Fabian M. Suchanek. Yago3: A knowledge base from multilingual wikipedias. In *CIDR*, 2014.

[30] Dai Quoc Nguyen, Tu Dinh Nguyen, Dat Quoc Nguyen, and Dinh Q. Phung. A novel embedding model for knowledge base completion based on convolutional neural network. In *NAACL-HLT*, 2018.

[31] Yankai Lin, Zhiyuan Liu, Maosong Sun, Yang Liu, and Xuan Zhu. Learning entity and relation embeddings for knowledge graph completion. In *AAAI*, 2015.

[32] Maximilian Nickel, Lorenzo Rosasco, and Tomaso A. Poggio. Holographic embeddings of knowledge graphs. In *AAAI*, 2016.

[33] Zhen Wang, Jianwen Zhang, Jianlin Feng, and Zheng Chen. Knowledge graph embedding by translating on hyperplanes. In *AAAI*, 2014.

[34] Guoliang Ji, Shizhu He, Liheng Xu, Kang Liu, and Jian Zhao. Knowledge graph embedding via dynamic mapping matrix. In *ACL*, 2015.

[35] Ni Lao, Tom Michael Mitchell, and William W. Cohen. Random walk inference and learning in a large scale knowledge base. In *EMNLP*, 2011.

[36] Matt Gardner, Partha Pratim Talukdar, Jayant Krishnamurthy, and Tom Michael Mitchell. Incorporating vector space similarity in random walk inference over knowledge bases. In *EMNLP*, 2014.

[37] Kelvin Guu, John Miller, and Percy S. Liang. Traversing knowledge graphs in vector space. In *EMNLP*, 2015.

[38] Yankai Lin, Zhiyuan Liu, and Maosong Sun. Modeling relation paths for representation learning of knowledge bases. In *EMNLP*, 2015.

[39] Kristina Toutanova, Victoria Lin, Wen tau Yih, Hoifung Poon, and Chris Quirk. Compositional learning of embeddings for relation paths in knowledge base and text. In *ACL*, 2016.

[40] Rajarshi Das, Arvind Neelakantan, David Belanger, and Andrew McCallum. Chains of reasoning over entities, relations, and text using recurrent neural networks. In *EACL*, 2017.

[41] Chen Liang, Jonathan Berant, Quoc V. Le, Kenneth D. Forbus, and Ni Lao. Neural symbolic machines: Learning semantic parsers on freebase with weak supervision. In *ACL*, 2016.

[42] Kenneth H. Craik. The nature of explanation. 1952.

[43] Brenden M. Lake, Tomer D. Ullman, Joshua B. Tenenbaum, and Samuel J Gershman. Building machines that learn and think like people. *The Behavioral and brain sciences*, 40:e253, 2017.

[44] Joan Bruna, Wojciech Zaremba, Arthur Szlam, and Yann LeCun. Spectral networks and locally connected networks on graphs. *CoRR*, abs/1312.6203, 2014.

[45] Mikael Henaff, Joan Bruna, and Yann LeCun. Deep convolutional networks on graph-structured data. *CoRR*, abs/1506.05163, 2015.

[46] David K. Duvenaud, Dougal Maclaurin, Jorge Aguilera-Iparraguirre, Rafael Gómez-Bombarelli, Timothy Hirzel, Alán Aspuru-Guzik, and Ryan P. Adams. Convolutional networks on graphs for learning molecular fingerprints. In *NIPS*, 2015.

[47] Steven M. Kearnes, Kevin McCloskey, Marc Berndl, Vijay S. Pande, and Patrick Riley. Molecular graph convolutions: moving beyond fingerprints. *Journal of computer-aided molecular design*, 30 8:595–608, 2016.

[48] Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. In *NIPS*, 2016.

[49] Mathias Niepert, Mohammed Hassan Ahmed, and Konstantin Kutzkov. Learning convolutional neural networks for graphs. In *ICML*, 2016.

[50] Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *CoRR*, abs/1609.02907, 2017.

[51] Michael M. Bronstein, Joan Bruna, Yann LeCun, Arthur Szlam, and Pierre Vandergheynst. Geometric deep learning: Going beyond euclidean data. *IEEE Signal Processing Magazine*, 34:18–42, 2017.

[52] Yujia Li, Daniel Tarlow, Marc Brockschmidt, and Richard S. Zemel. Gated graph sequence neural networks. *CoRR*, abs/1511.05493, 2016.

[53] Adam Santoro, David Raposo, David G. T. Barrett, Mateusz Malinowski, Razvan Pascanu, Peter W. Battaglia, and Timothy P. Lillicrap. A simple neural network module for relational reasoning. In *NIPS*, 2017.

[54] Peter W. Battaglia, Razvan Pascanu, Matthew Lai, Danilo Jimenez Rezende, and Koray Kavukcuoglu. Interaction networks for learning about objects, relations and physics. In *NIPS*, 2016.

[55] Adam Santoro, Ryan Faulkner, David Raposo, Jack W. Rae, Mike Chrzanowski, Théophane Weber, Daan Wierstra, Oriol Vinyals, Razvan Pascanu, and Timothy P. Lillicrap. Relational recurrent neural networks. In *NeurIPS*, 2018.

[56] Rasmus Berg Palm, Ulrich Paquet, and Ole Winther. Recurrent relational networks. In *NeurIPS*, 2018.

[57] Nicholas Watters, Daniel Zoran, Théophane Weber, Peter W. Battaglia, Razvan Pascanu, and Andrea Tacchetti. Visual interaction networks: Learning a physics simulator from video. In *NIPS*, 2017.

[58] Xiaolong Wang, Ross B. Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7794–7803, 2018.

[59] Wouter Kool. Attention solves your tsp , approximately. 2018.

[60] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *CoRR*, abs/1409.0473, 2015.

[61] Zhouhan Lin, Minwei Feng, Cícero Nogueira dos Santos, Mo Yu, Bing Xiang, Bowen Zhou, and Yoshua Bengio. A structured self-attentive sentence embedding. *CoRR*, abs/1703.03130, 2017.

[62] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NIPS*, 2017.

[63] Xiaoran Xu, Songpeng Zu, Chengliang Gao, Yuan Zhang, and Wei Feng. Modeling attention flow on graphs. *CoRR*, abs/1811.00497, 2018.

Appendix

7 Proof

Proposition. *Given a graph $\mathcal{G}$ (undirected or directed in both directions), we assume the probability of the degree of an arbitrary node being less than or equal to d is larger than p, i.e., $P(\deg(v) \leq d) > p, \forall v \in V$. Considering a sequence of consecutively expanding subgraphs $(G^0, G^1, \ldots, G^T)$, starting with $G^0 = \{v\}$, for all $t \geq 1$, we can ensure*

$$P\big(|V_{G^t}| \leq \frac{d(d-1)^t - 2}{d-2}\big) > p^{\frac{d(d-1)^{t-1}-2}{d-2}}. \tag{6}$$

Proof. We consider the extreme case of greedy consecutive expansion, where $G^t = G^{t-1} \cup \Delta G^t = G^{t-1} \cup \partial G^{t-1}$, since if this case satisfies the inequality, any case of consecutive expansion can also satisfy it. By definition, all the subgraphs G^t are a connected graph. Here, we use ΔV^t to denote $V_{\Delta G^t}$ for short. In the extreme case, we can ensure that the newly added nodes ΔV^t at step t only belong to the neighborhood of the last added nodes ΔV^{t-1}. Since for $t \geq 2$ each node in ΔV^{t-1} already has at least one edge within G^{t-1} due to the definition of connected graphs, we can have

$$P\big(|\Delta V^t| \leq |\Delta V^{t-1}|(d-1)\big) > p^{|\Delta V^{t-1}|}. \tag{7}$$

For $t = 1$, we have $P(|\Delta V^1| \leq d) > p$ and thus

$$P\big(|V_{G^1}| \leq 1 + d\big) > p. \tag{8}$$

For $t \geq 2$, based on $|V_{G^t}| = 1 + |\Delta V^1| + \ldots + |\Delta V^t|$, we obtain

$$P\big(|V_{G^t}| \leq 1 + d + d(d-1) + \ldots + d(d-1)^{t-1}\big) > p^{1+d+d(d-1)+\ldots+d(d-1)^{t-2}}, \tag{9}$$

which is

$$P\big(|V_{G^t}| \leq \frac{d(d-1)^t - 2}{d-2}\big) > p^{\frac{d(d-1)^{t-1}-2}{d-2}}. \tag{10}$$

We can find that $t = 1$ also satisfies this inequality. □

8 Hyperparameter Settings

Table 3: Our standard hyperparameter settings we use for each dataset plus their one-epoch training time. For experimental analysis, we only adjust one hyperparameter and keep the remaining fixed as the standard setting. For NELL995, the one-epoch training time means the average time cost of the 12 single-query-relation tasks.

Hyperparameter	FB15K-237	FB15K	WN18RR	WN18	YAGO3-10	NELL995
batch_size	80	80	100	100	100	10
n_dims_att	50	50	50	50	50	200
n_dims	100	100	100	100	100	200
max_sampling_per_step (in IGNN)	10000	10000	10000	10000	10000	10000
max_attending_from_per_step	20	20	20	20	20	100
max_sampling_per_node (in AGNN)	200	200	200	200	200	1000
max_attending_to_per_step	200	200	200	200	200	1000
n_steps_in_IGNN	2	1	2	1	1	1
n_steps_in_AGNN	6	6	8	8	6	5
learning_rate	0.001	0.001	0.001	0.001	0.0001	0.001
optimizer	Adam	Adam	Adam	Adam	Adam	Adam
grad_clipnorm	1	1	1	1	1	1
n_epochs	1	1	1	1	1	3
One-epoch training time (h)	25.7	63.7	4.3	8.5	185.0	0.12

The hyperparameters can be categorized into three groups:

- Normal hyperparameters, including *batch_size*, *n_dims_att*, *n_dims*, *learning_rate*, *grad_clipnorm*, and *n_epochs*. We set smaller dimensions, *n_dims_att*, for computation in the attention module, as it uses more edges than the message passing uses in AGNN, and also intuitively, it does not need to propagate high-dimensional messages but only compute scalar scores over a sampled neighborhood, in concert with the idea in the key-value mechanism [1]. We set $n_epochs = 1$ in most cases, indicating that our model can be trained well by one epoch only due to its fast convergence.
- The hyperparameters in charge of the sampling-attending horizon, including *max_sampling_per_step* that controls the maximum number to sample edges per step in IGNN, and *max_sampling_per_node*, *max_attending_from_per_step* and *max_attending_to_per_step* that control the maximum number to sample neighbors of each selected node per step per input, the maximum number of selected nodes for attending-from per step per input, and the maximum number of selected nodes in a sampled neighborhood for attending-to per step per input in AGNN.
- The hyperparameters in charge of the searching horizon, including *n_steps_in_IGNN* representing the number of propagation steps to run standard message passing in IGNN, and *n_steps_in_AGNN* representing the number of propagation steps to run pruned message passing in AGNN.

Note that we tune these hyperparameters according to not only their performances but also the computation resources available to us. In some cases, to deal with a very large knowledge graph with limited resources, we need to make a trade-off between efficiency and effectiveness. For example, each of NELL995's single-query-relation tasks has a small training set, though still with a large graph, so we can reduce the batch size in favor of affording larger dimensions and a larger sampling-attending horizon without any concern for waiting too long to finish one epoch.

9 More Experimental Results

Table 4: Comparison results on the FB15K and WN18 datasets. Results of [♠] are taken from [32], [♣] from [8], [♢] from [10], [♡] from [19], and [✠] from [11]. Our results take the form of "mean (std)".

	FB15K				**WN18**			
Metric (%)	H@1	H@3	H@10	MRR	H@1	H@3	H@10	MRR
TransE [♠]	29.7	57.8	74.9	46.3	11.3	88.8	94.3	49.5
HolE [♠]	40.2	61.3	73.9	52.4	93.0	94.5	94.9	93.8
DistMult [♣]	54.6	73.3	82.4	65.4	72.8	91.4	93.6	82.2
ComplEx [♣]	59.9	75.9	84.0	69.2	93.6	93.6	94.7	94.1
ConvE [♣]	55.8	72.3	83.1	65.7	93.5	94.6	95.6	94.3
RotatE [♢]	**74.6**	**83.0**	88.4	79.7	**94.4**	**95.2**	95.9	94.9
ComplEx-N3 [✠]	-	-	**91**	**86**	-	-	96	95
NeuralLP [♡]	-	-	83.7	76	-	-	94.5	94
DPMPN	72.6 (.4)	78.4 (.4)	83.4 (.5)	76.4 (.4)	91.6 (.8)	93.6 (.4)	94.9 (.4)	92.8 (.6)

Table 5: Comparison results on the YAGO3-10 dataset. Results of [♠] are taken from [8], [♣] from [11], and [✠] from [11].

	YAGO3-10			
Metric (%)	H@1	H@3	H@10	MRR
DistMult [♠]	24	38	54	34
ComplEx [♠]	26	40	55	36
ConvE [♠]	35	49	62	44
ComplEx-N3 [✠]	-	-	**71**	**58**
DPMPN	**48.4**	**59.5**	67.9	55.3

Table 6: Comparison results of MAP scores (%) on NELL995's single-query-relation KBC tasks. We take our baselines' results from [15]. No reports found on the last two in the paper.

Tasks	**NeuCFlow**	M-Walk	MINERVA	DeepPath	TransE	TransR
AthletePlaysForTeam	83.9 (0.5)	**84.7 (1.3)**	82.7 (0.8)	72.1 (1.2)	62.7	67.3
AthletePlaysInLeague	97.5 (0.1)	**97.8 (0.2)**	95.2 (0.8)	92.7 (5.3)	77.3	91.2
AthleteHomeStadium	**93.6 (0.1)**	91.9 (0.1)	92.8 (0.1)	84.6 (0.8)	71.8	72.2
AthletePlaysSport	**98.6 (0.0)**	98.3 (0.1)	**98.6 (0.1)**	91.7 (4.1)	87.6	96.3
TeamPlayssport	**90.4 (0.4)**	88.4 (1.8)	87.5 (0.5)	69.6 (6.7)	76.1	81.4
OrgHeadQuarteredInCity	94.7 (0.3)	**95.0 (0.7)**	94.5 (0.3)	79.0 (0.0)	62.0	65.7
WorksFor	**86.8 (0.0)**	84.2 (0.6)	82.7 (0.5)	69.9 (0.3)	67.7	69.2
PersonBornInLocation	**84.1 (0.5)**	81.2 (0.0)	78.2 (0.0)	75.5 (0.5)	71.2	81.2
PersonLeadsOrg	88.4 (0.1)	**88.8 (0.5)**	83.0 (2.6)	79.0 (1.0)	75.1	77.2
OrgHiredPerson	84.7 (0.8)	**88.8 (0.6)**	87.0 (0.3)	73.8 (1.9)	71.9	73.7
AgentBelongsToOrg	**89.3 (1.2)**	-	-	-	-	-
TeamPlaysInLeague	**97.2 (0.3)**	-	-	-	-	-

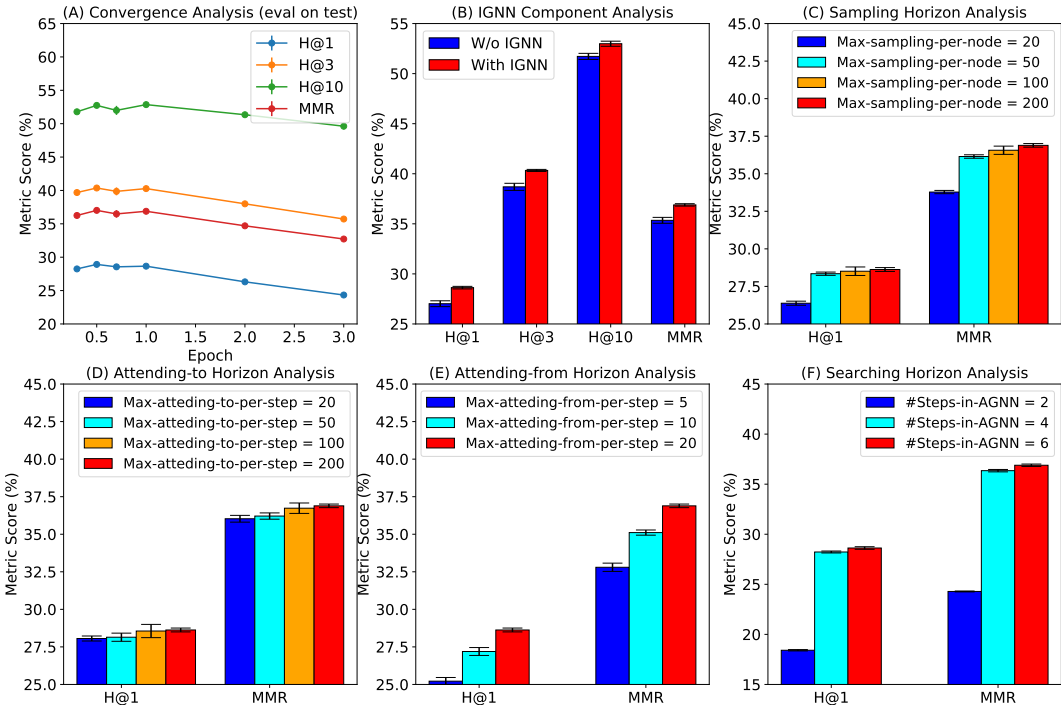


Figure 7: Experimental analysis on FB15K-237. (A) Convergence analysis: we pick six model snapshots at time points of 0.3, 0.5, 0.7, 1, 2, and 3 epochs during training and evaluate them on test; (B) IGNN component analysis: *w/o IGNN* uses zero step to run message passing, while *with IGNN* uses two steps; (C)-(F) Sampling, attending-to, attending-from and searching horizon analysis.

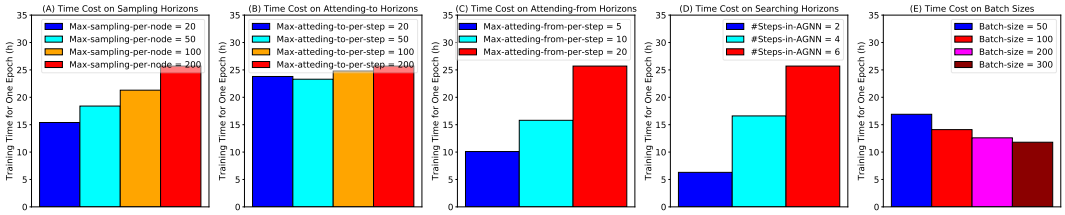


Figure 8: Analysis of time cost on FB15K-237: (A)-(D) measure the one-epoch training time on different horizon settings corresponding to Figure 7(C)-(F); (E) measures on different batch sizes using horizon setting *Max-sampled-edges-per-node*=20, *Max-seen-nodes-per-step*=20, *Max-attended-nodes-per-step*=20, and *#Steps-of-AGNN*=6.

10 More Visualization Resutls

10.1 Case study on the AthletePlaysForTeam task

In the case shown in Figure 9, the query is (*concept_personnorthamerica_michael_turner*, *concept:athleteplays-forteam*, ?) and a true answer is *concept_sportsteam_falcons*. From Figure 9, we can see our model learns that (*concept_personnorthamerica_michael_turner*, *concept:athletehomestadium*, *concept_stadiumoreventvenue_georgia_dome*) and (*concept_stadiumoreventvenue_georgia_dome*, *concept:teamhomestadium_inv*, *concept_sportsteam_falcons*) are two important facts to support the answer of *concept_sportsteam_falcons*. Besides, other facts, such as (*concept_athlete_joey_harrington*, *concept:athletehomestadium*, *concept_stadiumoreventvenue_georgia_dome*) and (*concept_athlete_joey_harrington*, *concept:athleteplaysforteam*, *concept_sportsteam_falcons*), provide a vivid example that a person or an athlete with *concept_stadiumoreventvenue_georgia_dome* as his or her home stadium might play for the team *concept_sportsteam_falcons*. We have such examples more than one, like *concept_athlete_roddy_white*'s and *concept_athlete_quarterback_matt_ryan*'s. The entity *concept_sportsleague_nfl* cannot help us differentiate the true answer from other NFL teams, but it can

at least exclude those non-NFL teams. In a word, our subgraph-structured representation can well capture the relational and compositional reasoning pattern.

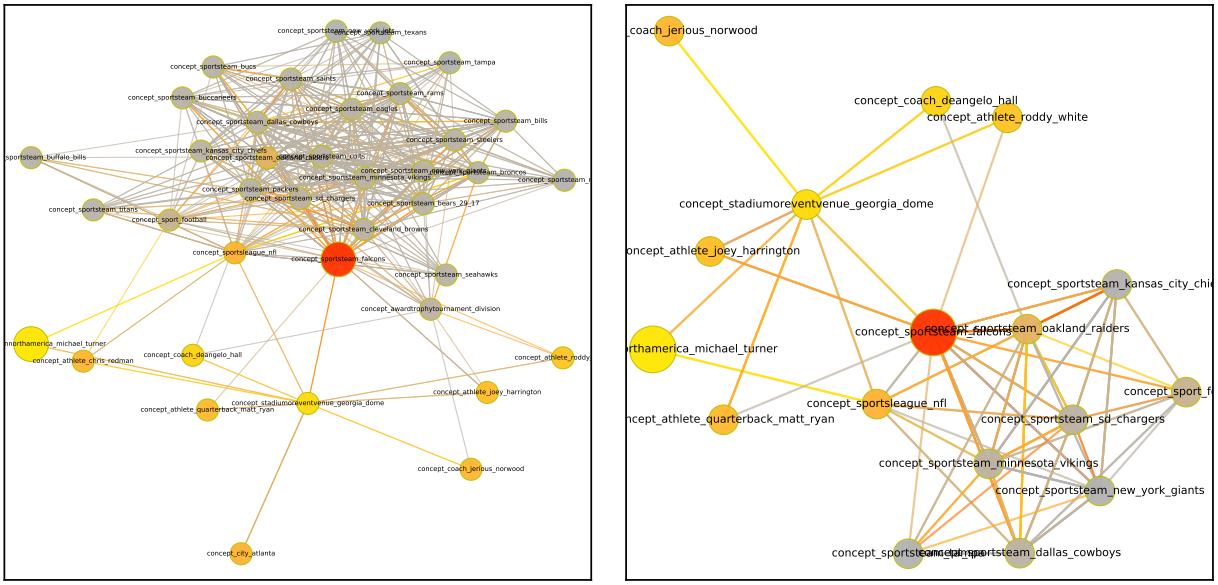


Figure 9: **AthletePlaysForTeam**. The head is *concept_personnorthamerica_michael_turner*, the query relation is *concept:athleteplaysforteam*, and the tail is *concept_sportsteam_falcons*. The left is a full subgraph derived with *max_attending_from_per_step*=20, and the right is a further pruned subgraph from the left based on attention. The big yellow node represents the head, and the big red node represents the tail. Color on the rest indicates attention scores over a T-step reasoning process, where grey means less attention, yellow means more attention gained during early steps, and red means gaining more attention when getting closer to the final step.

For the AthletePlaysForTeam task

Query: (concept_personnorthamerica_michael_turner, concept:athleteplaysforteam, concept_sportsteam_falcons)

Selected key edges:
concept_personnorthamerica_michael_turner, concept:agentbelongstoorganization, concept_sportsleague_nfl
concept_personnorthamerica_michael_turner, concept:athletehomestadium, concept_stadiumoreventvenue_georgia_dome
concept_sportsleague_nfl, concept:agentcompeteswithagent, concept_sportsleague_nfl
concept_sportsleague_nfl, concept:agentcompeteswithagent_inv, concept_sportsleague_nfl
concept_sportsleague_nfl, concept:teamplaysinleague_inv, concept_sportsteam_sd_chargers
concept_sportsleague_nfl, concept:leaguestadiums, concept_stadiumoreventvenue_georgia_dome
concept_sportsleague_nfl, concept:teamplaysinleague_inv, concept_sportsteam_falcons
concept_sportsleague_nfl, concept:agentbelongstoorganization_inv, concept_personnorthamerica_michael_turner
concept_stadiumoreventvenue_georgia_dome, concept:leaguestadiums_inv, concept_sportsleague_nfl
concept_stadiumoreventvenue_georgia_dome, concept:teamhomestadium_inv, concept_sportsteam_falcons
concept_stadiumoreventvenue_georgia_dome, concept:athletehomestadium_inv, concept_athlete_joey_harrington
concept_stadiumoreventvenue_georgia_dome, concept:athletehomestadium_inv, concept_athlete_roddy_white
concept_stadiumoreventvenue_georgia_dome, concept:athletehomestadium_inv, concept_coach_deangelo_hall
concept_stadiumoreventvenue_georgia_dome, concept:athletehomestadium_inv, concept_personnorthamerica_michael_turner
concept_sportsleague_nfl, concept:subpartoforganization_inv, concept_sportsteam_oakland_raiders
concept_sportsteam_sd_chargers, concept:teamplaysinleague, concept_sportsleague_nfl
concept_sportsteam_sd_chargers, concept:teamplaysagainstteam, concept_sportsteam_falcons
concept_sportsteam_sd_chargers, concept:teamplaysagainstteam_inv, concept_sportsteam_falcons
concept_sportsteam_sd_chargers, concept:teamplaysagainstteam, concept_sportsteam_oakland_raiders
concept_sportsteam_sd_chargers, concept:teamplaysagainstteam_inv, concept_sportsteam_oakland_raiders
concept_sportsteam_falcons, concept:teamplaysinleague, concept_sportsleague_nfl
concept_sportsteam_falcons, concept:teamplaysagainstteam, concept_sportsteam_sd_chargers
concept_sportsteam_falcons, concept:teamplaysagainstteam_inv, concept_sportsteam_sd_chargers
concept_sportsteam_falcons, concept:teamhomestadium, concept_stadiumoreventvenue_georgia_dome
concept_sportsteam_falcons, concept:teamplaysagainstteam, concept_sportsteam_oakland_raiders
concept_sportsteam_falcons, concept:teamplaysagainstteam_inv, concept_sportsteam_oakland_raiders
concept_sportsteam_falcons, concept:athleteledsportsteam_inv, concept_athlete_joey_harrington
concept_athlete_joey_harrington, concept:athletehomestadium, concept_stadiumoreventvenue_georgia_dome
concept_athlete_joey_harrington, concept:athleteledsportsteam, concept_sportsteam_falcons
concept_athlete_joey_harrington, concept:athleteplaysforteam, concept_sportsteam_falcons
concept_athlete_roddy_white, concept:athletehomestadium, concept_stadiumoreventvenue_georgia_dome
concept_athlete_roddy_white, concept:athleteplaysforteam, concept_sportsteam_falcons
concept_coach_deangelo_hall, concept:athletehomestadium, concept_stadiumoreventvenue_georgia_dome
concept_coach_deangelo_hall, concept:athleteplaysforteam, concept_sportsteam_oakland_raiders
concept_sportsleague_nfl, concept:teamplaysinleague_inv, concept_sportsteam_new_york_giants

concept_sportsteam_sd_chargers , concept:teamplaysagainstteam_inv , concept_sportsteam_new_york_giants
concept_sportsteam_falcons , concept:teamplaysagainstteam , concept_sportsteam_new_york_giants
concept_sportsteam_falcons , concept:teamplaysagainstteam_inv , concept_sportsteam_new_york_giants
concept_sportsteam_oakland_raiders , concept:teamplaysagainstteam_inv , concept_sportsteam_new_york_giants
concept_sportsteam_oakland_raiders , concept:teamplaysagainstteam , concept_sportsteam_sd_chargers
concept_sportsteam_oakland_raiders , concept:teamplaysagainstteam_inv , concept_sportsteam_sd_chargers
concept_sportsteam_oakland_raiders , concept:teamplaysagainstteam , concept_sportsteam_falcons
concept_sportsteam_oakland_raiders , concept:teamplaysagainstteam_inv , concept_sportsteam_falcons
concept_sportsteam_oakland_raiders , concept:agentcompeteswithagent , concept_sportsteam_oakland_raiders
concept_sportsteam_oakland_raiders , concept:agentcompeteswithagent_inv , concept_sportsteam_oakland_raiders
concept_sportsteam_new_york_giants , concept:teamplaysagainstteam , concept_sportsteam_sd_chargers
concept_sportsteam_new_york_giants , concept:teamplaysagainstteam , concept_sportsteam_falcons
concept_sportsteam_new_york_giants , concept:teamplaysagainstteam_inv , concept_sportsteam_falcons
concept_sportsteam_new_york_giants , concept:teamplaysagainstteam , concept_sportsteam_oakland_raiders

10.2 More results

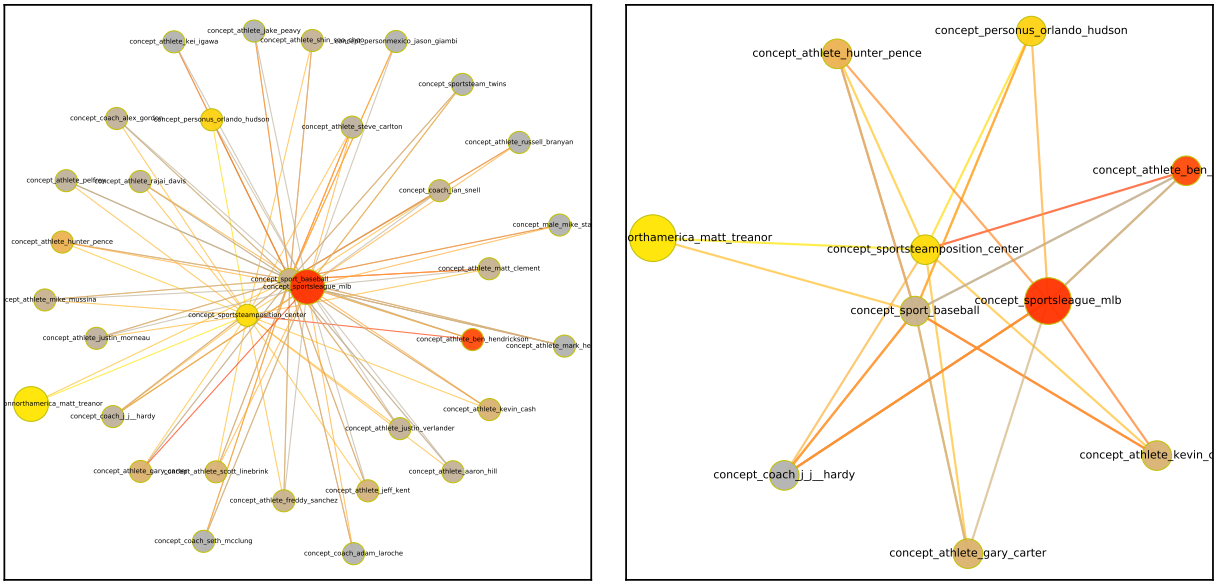


Figure 10: **AthletePlaysInLeague**. The head is *concept_personnorthamerica_matt_treanor*, the query relation is *concept:athleteplaysinleague*, and the tail is *concept_sportsleague_mlb*. The left is a full subgraph derived with *max_attending_from_per_step*=20, and the right is a further pruned subgraph from the left based on attention. The big yellow node represents the head, and the big red node represents the tail. Color on the rest indicates attention scores over a T-step reasoning process, where grey means less attention, yellow means more attention gained during early steps, and red means gaining more attention when getting closer to the final step.

For the AthletePlaysInLeague task

Query: (concept_personnorthamerica_matt_treanor , concept:athleteplaysinleague , concept_sportsleague_mlb)

Selected key edges:
concept_personnorthamerica_matt_treanor , concept:athleteflyouttosportsteamposition , concept_sportsteamposition_center
concept_personnorthamerica_matt_treanor , concept:athleteplayssport , concept_sport_baseball
concept_sportsteamposition_center , concept:athleteflyouttosportsteamposition_inv , concept_personus_orlando_hudson
concept_sportsteamposition_center , concept:athleteflyouttosportsteamposition_inv , concept_athlete_ben_hendrickson
concept_sportsteamposition_center , concept:athleteflyouttosportsteamposition_inv , concept_coach_j_j__hardy
concept_sportsteamposition_center , concept:athleteflyouttosportsteamposition_inv , concept_athlete_hunter_pence
concept_sport_baseball , concept:athleteplayssport_inv , concept_personus_orlando_hudson
concept_sport_baseball , concept:athleteplayssport_inv , concept_athlete_ben_hendrickson
concept_sport_baseball , concept:athleteplayssport_inv , concept_coach_j_j__hardy
concept_sport_baseball , concept:athleteplayssport_inv , concept_athlete_hunter_pence
concept_personus_orlando_hudson , concept:athleteplaysinleague , concept_sportsleague_mlb
concept_personus_orlando_hudson , concept:athleteplayssport , concept_sport_baseball
concept_athlete_ben_hendrickson , concept:coachesinleague , concept_sportsleague_mlb
concept_athlete_ben_hendrickson , concept:athleteplayssport , concept_sport_baseball
concept_coach_j_j__hardy , concept:coachesinleague , concept_sportsleague_mlb
concept_coach_j_j__hardy , concept:athleteplaysinleague , concept_sportsleague_mlb

concept_coach_j_j__hardy , concept:athleteplayssport , concept_sport_baseball
concept_athlete_hunter_pence , concept:athleteplaysinleague , concept_sportsleague_mlb
concept_athlete_hunter_pence , concept:athleteplayssport , concept_sport_baseball
concept_sportsleague_mlb , concept:coachesinleague_inv , concept_athlete_ben_hendrickson
concept_sportsleague_mlb , concept:coachesinleague_inv , concept_coach_j_j__hardy

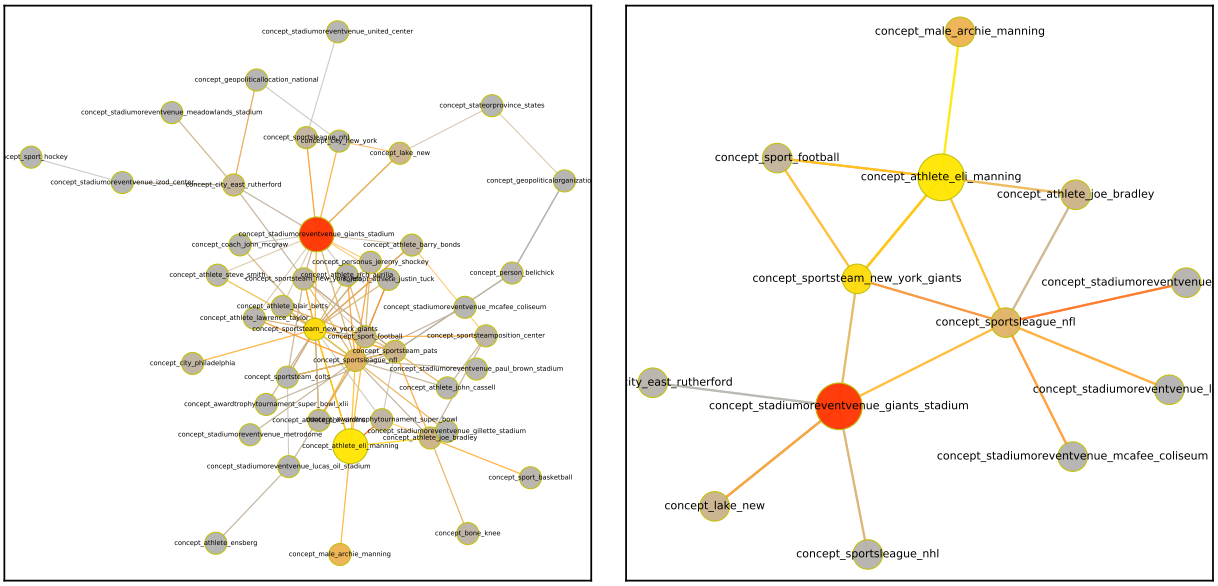


Figure 11: **AthleteHomeStadium**. The head is *concept_athlete_eli_manning*, the query relation is *concept:athletehomestadium*, and the tail is *concept_stadiumoreventvenue_giants_stadium*. The left is a full subgraph derived with *max_attending_from_per_step*=20, and the right is a further pruned subgraph from the left based on attention. The big yellow node represents the head, and the big red node represents the tail. Color on the rest indicates attention scores over a T-step reasoning process, where grey means less attention, yellow means more attention gained during early steps, and red means gaining more attention when getting closer to the final step.

For the AthleteHomeStadium task

Query: (concept_athlete_eli_manning , concept:athletehomestadium , concept_stadiumoreventvenue_giants_stadium)

Selected key edges:
concept_athlete_eli_manning , concept:personbelongstoorganization , concept_sportsteam_new_york_giants
concept_athlete_eli_manning , concept:athleteplaysforteam , concept_sportsteam_new_york_giants
concept_athlete_eli_manning , concept:athleteledsportsteam , concept_sportsteam_new_york_giants
concept_athlete_eli_manning , concept:athleteplaysinleague , concept_sportsleague_nfl
concept_athlete_eli_manning , concept:fatherofperson_inv , concept_male_archie_manning
concept_sportsteam_new_york_giants , concept:teamplaysinleague , concept_sportsleague_nfl
concept_sportsteam_new_york_giants , concept:teamhomestadium , concept_stadiumoreventvenue_giants_stadium
concept_sportsteam_new_york_giants , concept:personbelongstoorganization_inv , concept_athlete_eli_manning
concept_sportsteam_new_york_giants , concept:athleteplaysforteam_inv , concept_athlete_eli_manning
concept_sportsteam_new_york_giants , concept:athleteledsportsteam_inv , concept_athlete_eli_manning
concept_sportsleague_nfl , concept:teamplaysinleague_inv , concept_sportsteam_new_york_giants
concept_sportsleague_nfl , concept:agentcompeteswithagent , concept_sportsleague_nfl
concept_sportsleague_nfl , concept:agentcompeteswithagent_inv , concept_sportsleague_nfl
concept_sportsleague_nfl , concept:leaguestadiums , concept_stadiumoreventvenue_giants_stadium
concept_sportsleague_nfl , concept:athleteplaysinleague_inv , concept_athlete_eli_manning
concept_male_archie_manning , concept:fatherofperson , concept_athlete_eli_manning
concept_sportsleague_nfl , concept:leaguestadiums , concept_stadiumoreventvenue_paul_brown_stadium
concept_stadiumoreventvenue_giants_stadium , concept:teamhomestadium_inv , concept_sportsteam_new_york_giants
concept_stadiumoreventvenue_giants_stadium , concept:leaguestadiums_inv , concept_sportsleague_nfl
concept_stadiumoreventvenue_giants_stadium , concept:proxyfor_inv , concept_city_east_rutherford
concept_city_east_rutherford , concept:proxyfor , concept_stadiumoreventvenue_giants_stadium
concept_stadiumoreventvenue_paul_brown_stadium , concept:leaguestadiums_inv , concept_sportsleague_nfl

For the AthletePlaysSport task

Query: (concept_athlete_vernon_wells , concept:athleteplayssport , concept_sport_baseball)

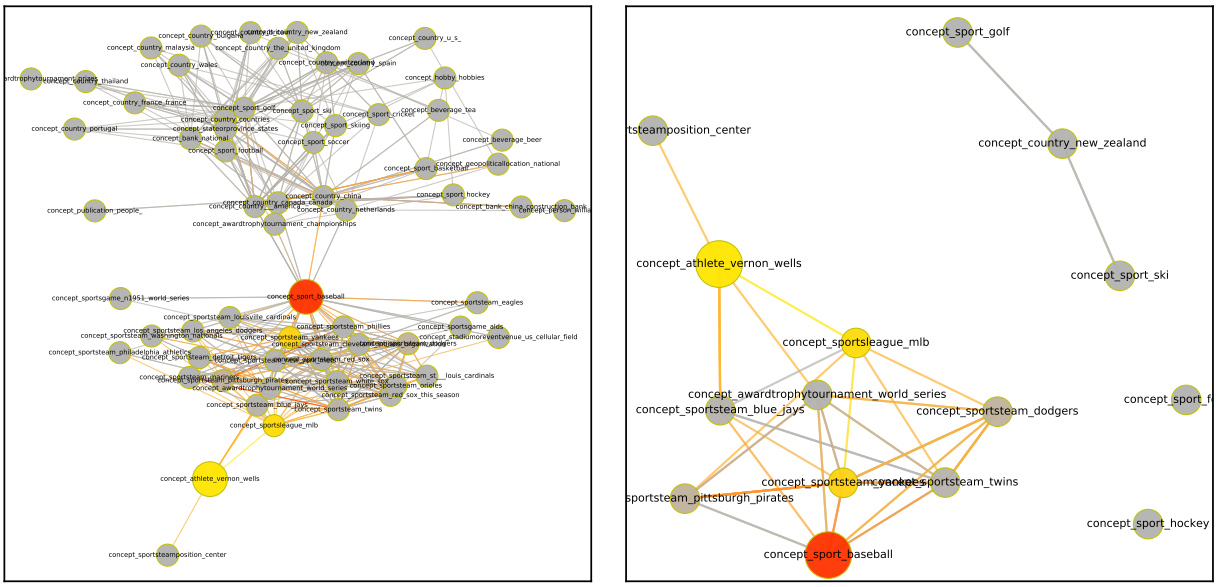


Figure 12: **AthletePlaysSport**. The head is *concept_athlete_vernon_wells*, the query relation is *concept:athleteplayssport*, and the tail is *concept_sport_baseball*. The left is a full subgraph derived with *max_attending_from_per_step*=20, and the right is a further pruned subgraph from the left based on attention. The big yellow node represents the head, and the big red node represents the tail. Color on the rest indicates attention scores over a T-step reasoning process, where grey means less attention, yellow means more attention gained during early steps, and red means gaining more attention when getting closer to the final step.

Selected key edges:
concept_athlete_vernon_wells , concept:athleteplaysinleague , concept_sportsleague_mlb
concept_athlete_vernon_wells , concept:coachwontrophy , concept_awardtrophytournament_world_series
concept_athlete_vernon_wells , concept:agentcollaborateswithagent_inv , concept_sportsteam_blue_jays
concept_athlete_vernon_wells , concept:personbelongstoorganization , concept_sportsteam_blue_jays
concept_athlete_vernon_wells , concept:athleteplaysforteam , concept_sportsteam_blue_jays
concept_athlete_vernon_wells , concept:athleteledsportsteam , concept_sportsteam_blue_jays
concept_sportsleague_mlb , concept:teamplaysinleague_inv , concept_sportsteam_dodgers
concept_sportsleague_mlb , concept:teamplaysinleague_inv , concept_sportsteam_yankees
concept_sportsleague_mlb , concept:teamplaysinleague_inv , concept_sportsteam_pittsburgh_pirates
concept_awardtrophytournament_world_series , concept:teamwontrophy_inv , concept_sportsteam_dodgers
concept_awardtrophytournament_world_series , concept:teamwontrophy_inv , concept_sportsteam_yankees
concept_awardtrophytournament_world_series , concept:awardtrophytournamentisthechampionshipgameofthenationalsport ,
concept_sport_baseball
concept_awardtrophytournament_world_series , concept:teamwontrophy_inv , concept_sportsteam_pittsburgh_pirates
concept_sportsteam_blue_jays , concept:teamplaysinleague , concept_sportsleague_mlb
concept_sportsteam_blue_jays , concept:teamplaysagainstteam , concept_sportsteam_yankees
concept_sportsteam_blue_jays , concept:teamplayssport , concept_sport_baseball
concept_sportsteam_dodgers , concept:teamplaysagainstteam , concept_sportsteam_yankees
concept_sportsteam_dodgers , concept:teamplaysagainstteam_inv , concept_sportsteam_yankees
concept_sportsteam_dodgers , concept:teamwontrophy , concept_awardtrophytournament_world_series
concept_sportsteam_dodgers , concept:teamplayssport , concept_sport_baseball
concept_sportsteam_yankees , concept:teamplaysagainstteam , concept_sportsteam_dodgers
concept_sportsteam_yankees , concept:teamplaysagainstteam_inv , concept_sportsteam_dodgers
concept_sportsteam_yankees , concept:teamwontrophy , concept_awardtrophytournament_world_series
concept_sportsteam_yankees , concept:teamplayssport , concept_sport_baseball
concept_sportsteam_yankees , concept:teamplaysagainstteam , concept_sportsteam_pittsburgh_pirates
concept_sportsteam_yankees , concept:teamplaysagainstteam_inv , concept_sportsteam_pittsburgh_pirates
concept_sport_baseball , concept:teamplayssport_inv , concept_sportsteam_dodgers
concept_sport_baseball , concept:teamplayssport_inv , concept_sportsteam_yankees
concept_sport_baseball , concept:awardtrophytournamentisthechampionshipgameofthenationalsport_inv ,
concept_awardtrophytournament_world_series
concept_sport_baseball , concept:teamplayssport_inv , concept_sportsteam_pittsburgh_pirates
concept_sportsteam_pittsburgh_pirates , concept:teamplaysagainstteam , concept_sportsteam_yankees
concept_sportsteam_pittsburgh_pirates , concept:teamplaysagainstteam_inv , concept_sportsteam_yankees
concept_sportsteam_pittsburgh_pirates , concept:teamwontrophy , concept_awardtrophytournament_world_series
concept_sportsteam_pittsburgh_pirates , concept:teamplayssport , concept_sport_baseball

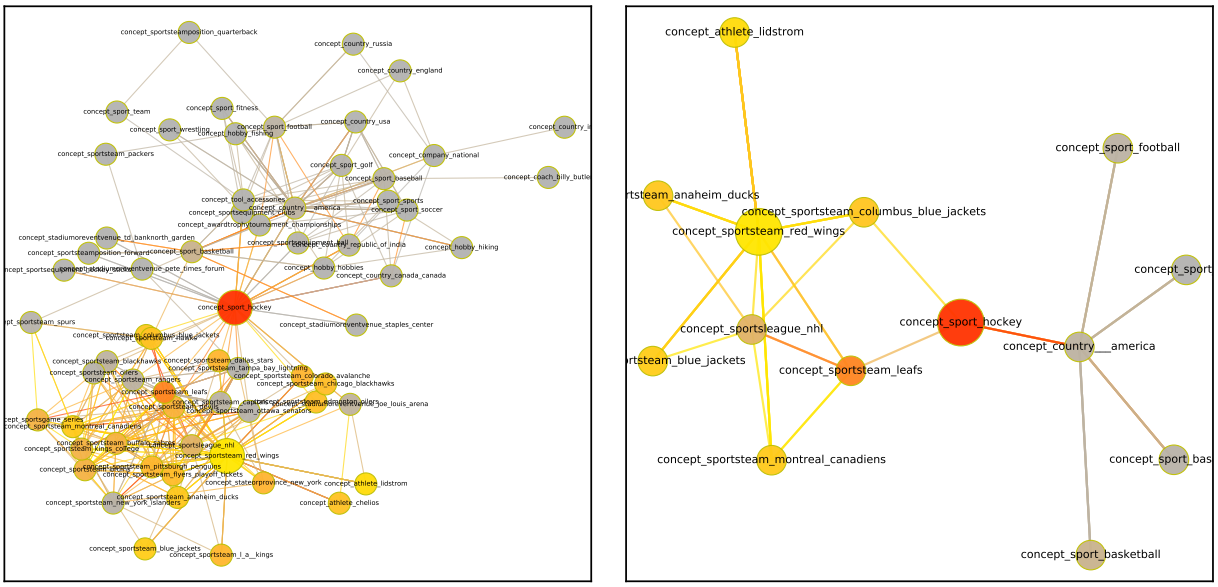


Figure 13: **TeamPlaysSport**. The head is *concept_sportsteam_red_wings*, the query relation is *concept:teamplayssport*, and the tail is *concept_sport_hockey*. The left is a full subgraph derived with *max_attending_from_per_step*=20, and the right is a further pruned subgraph from the left based on attention. The big yellow node represents the head, and the big red node represents the tail. Color on the rest indicates attention scores over a T-step reasoning process, where grey means less attention, yellow means more attention gained during early steps, and red means gaining more attention when getting closer to the final step.

For the TeamPlaysSport task

```
Query: (concept_sportsteam_red_wings, concept:teamplayssport, concept_sport_hockey)

Selected key edges:
concept_sportsteam_red_wings, concept:teamplaysagainstteam, concept_sportsteam_montreal_canadiens
concept_sportsteam_red_wings, concept:teamplaysagainstteam_inv, concept_sportsteam_montreal_canadiens
concept_sportsteam_red_wings, concept:teamplaysagainstteam, concept_sportsteam_blue_jackets
concept_sportsteam_red_wings, concept:teamplaysagainstteam_inv, concept_sportsteam_blue_jackets
concept_sportsteam_red_wings, concept:worksfor_inv, concept_athlete_lidstrom
concept_sportsteam_red_wings, concept:organizationhiredperson, concept_athlete_lidstrom
concept_sportsteam_red_wings, concept:athleteplaysforteam_inv, concept_athlete_lidstrom
concept_sportsteam_red_wings, concept:athleteledsportsteam_inv, concept_athlete_lidstrom
concept_sportsteam_montreal_canadiens, concept:teamplaysagainstteam, concept_sportsteam_red_wings
concept_sportsteam_montreal_canadiens, concept:teamplaysagainstteam_inv, concept_sportsteam_red_wings
concept_sportsteam_montreal_canadiens, concept:teamplaysinleague, concept_sportsleague_nhl
concept_sportsteam_montreal_canadiens, concept:teamplaysagainstteam, concept_sportsteam_leafs
concept_sportsteam_montreal_canadiens, concept:teamplaysagainstteam_inv, concept_sportsteam_leafs
concept_sportsteam_blue_jackets, concept:teamplaysagainstteam, concept_sportsteam_red_wings
concept_sportsteam_blue_jackets, concept:teamplaysagainstteam_inv, concept_sportsteam_red_wings
concept_sportsteam_blue_jackets, concept:teamplaysinleague, concept_sportsleague_nhl
concept_athlete_lidstrom, concept:worksfor, concept_sportsteam_red_wings
concept_athlete_lidstrom, concept:organizationhiredperson_inv, concept_sportsteam_red_wings
concept_athlete_lidstrom, concept:athleteplaysforteam, concept_sportsteam_red_wings
concept_athlete_lidstrom, concept:athleteledsportsteam, concept_sportsteam_red_wings
concept_sportsteam_red_wings, concept:teamplaysinleague, concept_sportsleague_nhl
concept_sportsteam_red_wings, concept:teamplaysagainstteam, concept_sportsteam_leafs
concept_sportsteam_red_wings, concept:teamplaysagainstteam_inv, concept_sportsteam_leafs
concept_sportsleague_nhl, concept:agentcompeteswithagent, concept_sportsleague_nhl
concept_sportsleague_nhl, concept:agentcompeteswithagent_inv, concept_sportsleague_nhl
concept_sportsleague_nhl, concept:teamplaysinleague_inv, concept_sportsteam_leafs
concept_sportsteam_leafs, concept:teamplaysinleague, concept_sportsleague_nhl
concept_sportsteam_leafs, concept:teamplayssport, concept_sport_hockey
```

For the OrganizationHeadQuarteredInCity task

```
Query: (concept_company_disney, concept:organizationheadquarteredincity, concept_city_burbank)

Selected key edges:
```

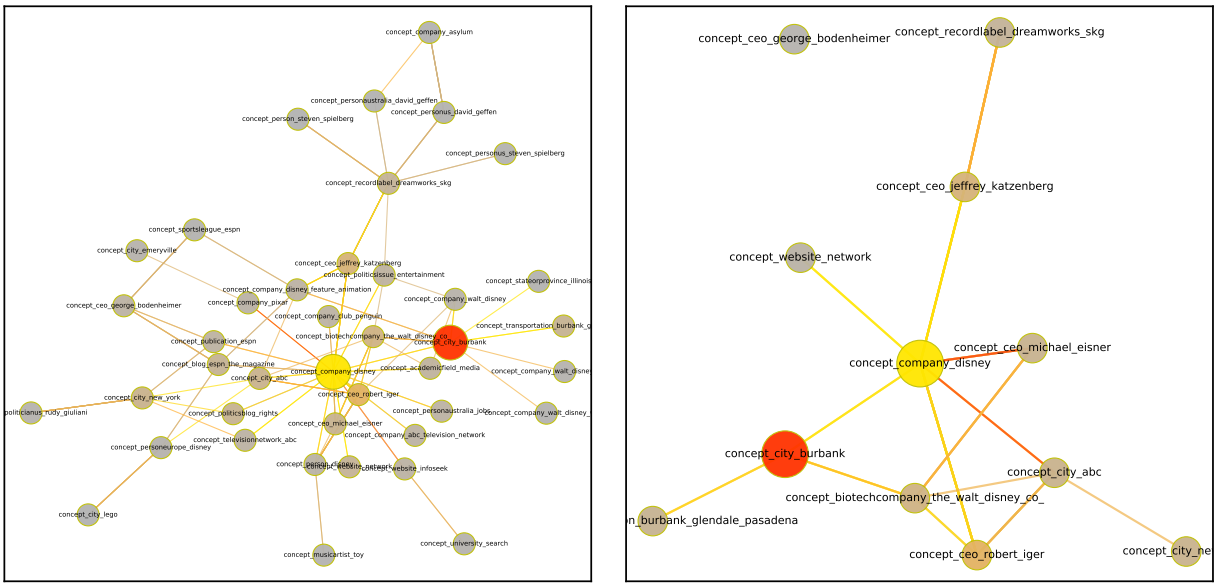


Figure 14: **OrganizationHeadQuarteredInCity**. The head is *concept_company_disney*, the query relation is *concept:organizationheadquarteredincity*, and the tail is *concept_city_burbank*. The left is a full subgraph derived with *max_attending_from_per_step*=20, and the right is a further pruned subgraph from the left based on attention. The big yellow node represents the head, and the big red node represents the tail. Color on the rest indicates attention scores over a T-step reasoning process, where grey means less attention, yellow means more attention gained during early steps, and red means gaining more attention when getting closer to the final step.

concept_company_disney , concept:headquarteredin , concept_city_burbank
concept_company_disney , concept:subpartoforganization_inv , concept_website_network
concept_company_disney , concept:worksfor_inv , concept_ceo_robert_iger
concept_company_disney , concept:proxyfor_inv , concept_ceo_robert_iger
concept_company_disney , concept:personleadsorganization_inv , concept_ceo_robert_iger
concept_company_disney , concept:ceoof_inv , concept_ceo_robert_iger
concept_company_disney , concept:personleadsorganization_inv , concept_ceo_jeffrey_katzenberg
concept_company_disney , concept:organizationhiredperson , concept_ceo_jeffrey_katzenberg
concept_company_disney , concept:organizationterminatedperson , concept_ceo_jeffrey_katzenberg
concept_city_burbank , concept:headquarteredin_inv , concept_company_disney
concept_city_burbank , concept:headquarteredin_inv , concept_biotechcompany_the_walt_disney_co_
concept_website_network , concept:subpartoforganization , concept_company_disney
concept_ceo_robert_iger , concept:worksfor , concept_company_disney
concept_ceo_robert_iger , concept:proxyfor , concept_company_disney
concept_ceo_robert_iger , concept:personleadsorganization , concept_company_disney
concept_ceo_robert_iger , concept:ceoof , concept_company_disney
concept_ceo_robert_iger , concept:topmemberoforganization , concept_biotechcompany_the_walt_disney_co_
concept_ceo_robert_iger , concept:organizationterminatedperson_inv , concept_biotechcompany_the_walt_disney_co_
concept_ceo_jeffrey_katzenberg , concept:personleadsorganization , concept_company_disney
concept_ceo_jeffrey_katzenberg , concept:organizationhiredperson_inv , concept_company_disney
concept_ceo_jeffrey_katzenberg , concept:organizationterminatedperson_inv , concept_company_disney
concept_ceo_jeffrey_katzenberg , concept:worksfor , concept_recordlabel_dreamworks_skg
concept_ceo_jeffrey_katzenberg , concept:topmemberoforganization , concept_recordlabel_dreamworks_skg
concept_ceo_jeffrey_katzenberg , concept:organizationterminatedperson_inv , concept_recordlabel_dreamworks_skg
concept_ceo_jeffrey_katzenberg , concept:ceoof , concept_recordlabel_dreamworks_skg
concept_biotechcompany_the_walt_disney_co_ , concept:headquarteredin , concept_city_burbank
concept_biotechcompany_the_walt_disney_co_ , concept:organizationheadquarteredincity , concept_city_burbank
concept_recordlabel_dreamworks_skg , concept:worksfor_inv , concept_ceo_jeffrey_katzenberg
concept_recordlabel_dreamworks_skg , concept:topmemberoforganization_inv , concept_ceo_jeffrey_katzenberg
concept_recordlabel_dreamworks_skg , concept:organizationterminatedperson , concept_ceo_jeffrey_katzenberg
concept_recordlabel_dreamworks_skg , concept:ceoof_inv , concept_ceo_jeffrey_katzenberg
concept_city_burbank , concept:airportincity_inv , concept_transportation_burbank_glendale_pasadena
concept_transportation_burbank_glendale_pasadena , concept:airportincity , concept_city_burbank

For the WorksFor task

Query: (concept_scientist_balmer , concept:worksfor , concept_university_microsoft)

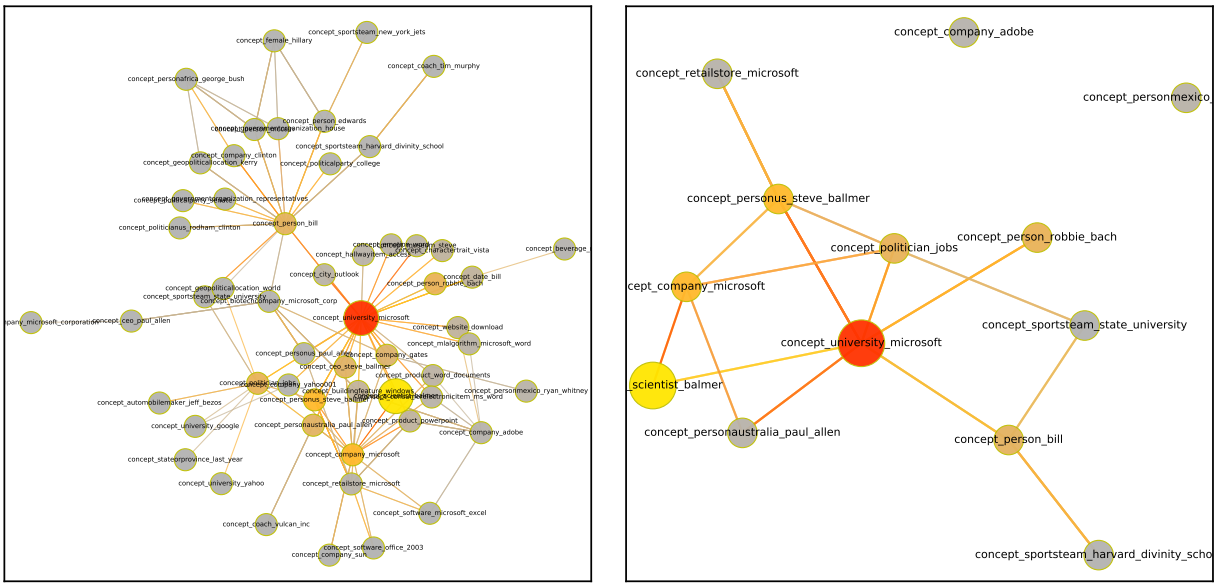


Figure 15: **WorksFor**. The head is *concept_scientist_balmer*, the query relation is *concept:worksfor*, and the tail is *concept_university_microsoft*. The left is a full subgraph derived with *max_attending_from_per_step*=20, and the right is a further pruned subgraph from the left based on attention. The big yellow node represents the head, and the big red node represents the tail. Color on the rest indicates attention scores over a T-step reasoning process, where grey means less attention, yellow means more attention gained during early steps, and red means gaining more attention when getting closer to the final step.

```
Selected key edges:
concept_scientist_balmer , concept:topmemberoforganization , concept_company_microsoft
concept_scientist_balmer , concept:organizationterminatedperson_inv , concept_university_microsoft
concept_company_microsoft , concept:topmemberoforganization_inv , concept_personus_steve_ballmer
concept_company_microsoft , concept:topmemberoforganization_inv , concept_scientist_balmer
concept_university_microsoft , concept:agentcollaborateswithagent , concept_personus_steve_ballmer
concept_university_microsoft , concept:personleadsorganization_inv , concept_personus_steve_ballmer
concept_university_microsoft , concept:personleadsorganization_inv , concept_person_bill
concept_university_microsoft , concept:organizationterminatedperson , concept_scientist_balmer
concept_university_microsoft , concept:personleadsorganization_inv , concept_person_robbie_bach
concept_personus_steve_ballmer , concept:topmemberoforganization , concept_company_microsoft
concept_personus_steve_ballmer , concept:agentcollaborateswithagent_inv , concept_university_microsoft
concept_personus_steve_ballmer , concept:personleadsorganization , concept_university_microsoft
concept_personus_steve_ballmer , concept:worksfor , concept_university_microsoft
concept_personus_steve_ballmer , concept:proxyfor , concept_retailstore_microsoft
concept_personus_steve_ballmer , concept:subpartof , concept_retailstore_microsoft
concept_personus_steve_ballmer , concept:agentcontrols , concept_retailstore_microsoft
concept_person_bill , concept:personleadsorganization , concept_university_microsoft
concept_person_bill , concept:worksfor , concept_university_microsoft
concept_person_robbie_bach , concept:personleadsorganization , concept_university_microsoft
concept_person_robbie_bach , concept:worksfor , concept_university_microsoft
concept_retailstore_microsoft , concept:proxyfor_inv , concept_personus_steve_ballmer
concept_retailstore_microsoft , concept:subpartof_inv , concept_personus_steve_ballmer
concept_retailstore_microsoft , concept:agentcontrols_inv , concept_personus_steve_ballmer
```

For the PersonBornInLocation task

```
Query: (concept_person_mark001 , concept:personborninlocation , concept_county_york_city)

Selected key edges:
concept_person_mark001 , concept:persongraduatedfromuniversity , concept_university_college
concept_person_mark001 , concept:persongraduatedschool , concept_university_college
concept_person_mark001 , concept:persongraduatedfromuniversity , concept_university_state_university
concept_person_mark001 , concept:persongraduatedschool , concept_university_state_university
concept_person_mark001 , concept:personbornincity , concept_city_hampshire
concept_person_mark001 , concept:hasspouse , concept_person_diane001
concept_person_mark001 , concept:hasspouse_inv , concept_person_diane001
concept_university_college , concept:persongraduatedfromuniversity_inv , concept_person_mark001
```

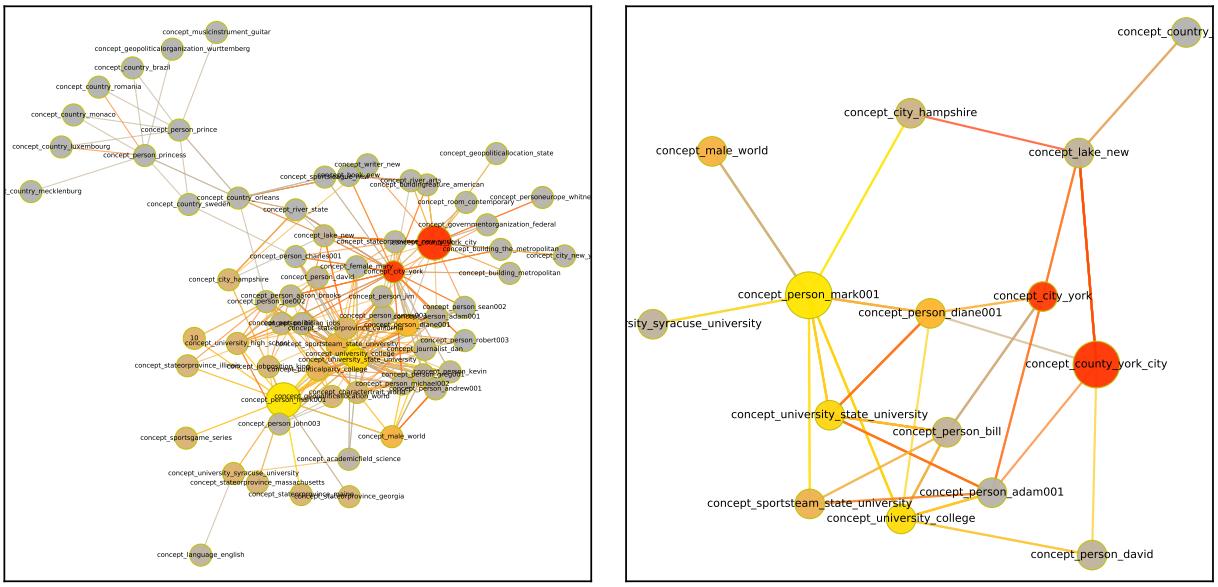


Figure 16: **PersonBornInLocation**. The head is *concept_person_mark001*, the query relation is *concept:personborninlocation*, and the tail is *concept_county_york_city*. The left is a full subgraph derived with *max_attending_from_per_step*=20, and the right is a further pruned subgraph from the left based on attention. The big yellow node represents the head, and the big red node represents the tail. Color on the rest indicates attention scores over a T-step reasoning process, where grey means less attention, yellow means more attention gained during early steps, and red means gaining more attention when getting closer to the final step.

concept_university_college , concept:persongraduatedschool_inv , concept_person_mark001
concept_university_college , concept:persongraduatedfromuniversity_inv , concept_person_bill
concept_university_college , concept:persongraduatedschool_inv , concept_person_bill
concept_university_state_university , concept:persongraduatedfromuniversity_inv , concept_person_mark001
concept_university_state_university , concept:persongraduatedschool_inv , concept_person_mark001
concept_university_state_university , concept:persongraduatedfromuniversity_inv , concept_person_bill
concept_university_state_university , concept:persongraduatedschool_inv , concept_person_bill
concept_city_hampshire , concept:personbornincity_inv , concept_person_mark001
concept_person_diane001 , concept:persongraduatedfromuniversity , concept_university_state_university
concept_person_diane001 , concept:persongraduatedschool , concept_university_state_university
concept_person_diane001 , concept:hasspouse , concept_person_mark001
concept_person_diane001 , concept:hasspouse_inv , concept_person_mark001
concept_person_diane001 , concept:personborninlocation , concept_county_york_city
concept_university_state_university , concept:persongraduatedfromuniversity_inv , concept_person_diane001
concept_university_state_university , concept:persongraduatedschool_inv , concept_person_diane001
concept_person_bill , concept:personbornincity , concept_city_york
concept_person_bill , concept:personborninlocation , concept_city_york
concept_person_bill , concept:persongraduatedfromuniversity , concept_university_college
concept_person_bill , concept:persongraduatedschool , concept_university_college
concept_person_bill , concept:persongraduatedfromuniversity , concept_university_state_university
concept_person_bill , concept:persongraduatedschool , concept_university_state_university
concept_city_york , concept:personbornincity_inv , concept_person_bill
concept_city_york , concept:personbornincity_inv , concept_person_diane001
concept_university_college , concept:persongraduatedfromuniversity_inv , concept_person_diane001
concept_person_diane001 , concept:personbornincity , concept_city_york

For the PersonLeadsOrganization task

Query: (concept_journalist_bill_plante , concept:personleadsorganization , concept_company_cnn__pbs)

Selected key edges:
concept_journalist_bill_plante , concept:worksfor , concept_televisionnetwork_cbs
concept_journalist_bill_plante , concept:agentcollaborateswithagent_inv , concept_televisionnetwork_cbs
concept_televisionnetwork_cbs , concept:worksfor_inv , concept_journalist_walter_cronkite
concept_televisionnetwork_cbs , concept:agentcollaborateswithagent , concept_journalist_walter_cronkite
concept_televisionnetwork_cbs , concept:worksfor_inv , concept_personus_scott_pelley
concept_televisionnetwork_cbs , concept:worksfor_inv , concept_actor_daniel_schorr
concept_televisionnetwork_cbs , concept:worksfor_inv , concept_person_edward_r__murrow

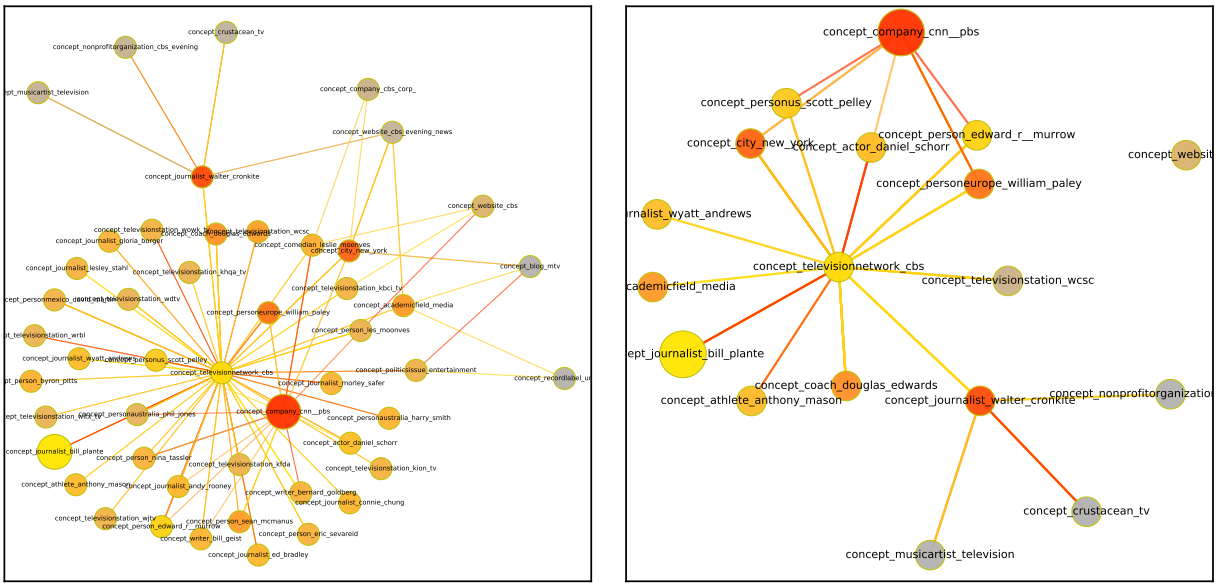


Figure 17: **PersonLeadsOrganization**. The head is *concept_journalist_bill_plante*, the query relation is *concept:organizationheadquarteredincity*, and the tail is *concept_company_cnn__pbs*. The left is a full subgraph derived with *max_attending_from_per_step*=20, and the right is a further pruned subgraph from the left based on attention. The big yellow node represents the head, and the big red node represents the tail. Color on the rest indicates attention scores over a T-step reasoning process, where grey means less attention, yellow means more attention gained during early steps, and red means gaining more attention when getting closer to the final step.

concept_televisionnetwork_cbs , concept:agentcollaborateswithagent , concept_person_edward_r__murrow
concept_televisionnetwork_cbs , concept:worksfor_inv , concept_journalist_bill_plante
concept_televisionnetwork_cbs , concept:agentcollaborateswithagent , concept_journalist_bill_plante
concept_journalist_walter_cronkite , concept:worksfor , concept_televisionnetwork_cbs
concept_journalist_walter_cronkite , concept:agentcollaborateswithagent_inv , concept_televisionnetwork_cbs
concept_journalist_walter_cronkite , concept:worksfor , concept_nonprofitorganization_cbs_evening
concept_personus_scott_pelley , concept:worksfor , concept_televisionnetwork_cbs
concept_personus_scott_pelley , concept:personleadsorganization , concept_televisionnetwork_cbs
concept_personus_scott_pelley , concept:personleadsorganization , concept_company_cnn__pbs
concept_actor_daniel_schorr , concept:worksfor , concept_televisionnetwork_cbs
concept_actor_daniel_schorr , concept:personleadsorganization , concept_televisionnetwork_cbs
concept_actor_daniel_schorr , concept:personleadsorganization , concept_company_cnn__pbs
concept_person_edward_r__murrow , concept:worksfor , concept_televisionnetwork_cbs
concept_person_edward_r__murrow , concept:agentcollaborateswithagent_inv , concept_televisionnetwork_cbs
concept_person_edward_r__murrow , concept:personleadsorganization , concept_televisionnetwork_cbs
concept_person_edward_r__murrow , concept:personleadsorganization , concept_company_cnn__pbs
concept_televisionnetwork_cbs , concept:organizationheadquarteredincity , concept_city_new_york
concept_televisionnetwork_cbs , concept:headquarteredin , concept_city_new_york
concept_televisionnetwork_cbs , concept:agentcollaborateswithagent , concept_personeurope_william_paley
concept_televisionnetwork_cbs , concept:topmemberoforganization_inv , concept_personeurope_william_paley
concept_company_cnn__pbs , concept:headquarteredin , concept_city_new_york
concept_company_cnn__pbs , concept:personbelongstoorganization_inv , concept_personeurope_william_paley
concept_nonprofitorganization_cbs_evening , concept:worksfor_inv , concept_journalist_walter_cronkite
concept_city_new_york , concept:organizationheadquarteredincity_inv , concept_televisionnetwork_cbs
concept_city_new_york , concept:headquarteredin_inv , concept_televisionnetwork_cbs
concept_city_new_york , concept:headquarteredin_inv , concept_company_cnn__pbs
concept_personeurope_william_paley , concept:agentcollaborateswithagent_inv , concept_televisionnetwork_cbs
concept_personeurope_william_paley , concept:topmemberoforganization , concept_televisionnetwork_cbs
concept_personeurope_william_paley , concept:personbelongstoorganization , concept_company_cnn__pbs
concept_personeurope_william_paley , concept:personleadsorganization , concept_company_cnn__pbs

For the OrganizationHiredPerson task

Query: (concept_stateorprovince_afternoon , concept:organizationhiredperson , concept_personmexico_ryan_whitney)

Selected key edges:
concept_stateorprovince_afternoon , concept:atdate , concept_dateliteral_n2007
concept_stateorprovince_afternoon , concept:atdate , concept_date_n2003

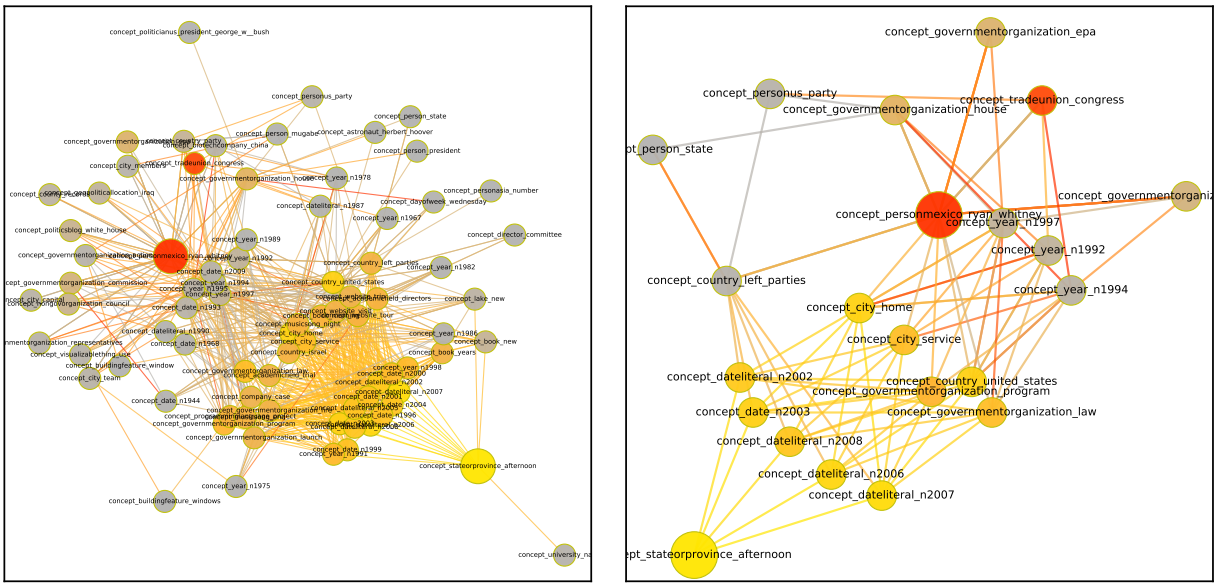


Figure 18: **OrganizationHiredPerson**. The head is *concept_stateorprovince_afternoon*, the query relation is *concept:organizationhiredperson*, and the tail is *concept_personmexico_ryan_whitney*. The left is a full subgraph derived with *max_attending_from_per_step*=20, and the right is a further pruned subgraph from the left based on attention. The big yellow node represents the head, and the big red node represents the tail. Color on the rest indicates attention scores over a T-step reasoning process, where grey means less attention, yellow means more attention gained during early steps, and red means gaining more attention when getting closer to the final step.

```
concept_stateorprovince_afternoon , concept:atdate , concept_dateliteral_n2006
concept_dateliteral_n2007 , concept:atdate_inv , concept_country_united_states
concept_dateliteral_n2007 , concept:atdate_inv , concept_city_home
concept_dateliteral_n2007 , concept:atdate_inv , concept_city_service
concept_dateliteral_n2007 , concept:atdate_inv , concept_country_left_parties
concept_date_n2003 , concept:atdate_inv , concept_country_united_states
concept_date_n2003 , concept:atdate_inv , concept_city_home
concept_date_n2003 , concept:atdate_inv , concept_city_service
concept_date_n2003 , concept:atdate_inv , concept_country_left_parties
concept_dateliteral_n2006 , concept:atdate_inv , concept_country_united_states
concept_dateliteral_n2006 , concept:atdate_inv , concept_city_home
concept_dateliteral_n2006 , concept:atdate_inv , concept_city_service
concept_dateliteral_n2006 , concept:atdate_inv , concept_country_left_parties
concept_country_united_states , concept:atdate , concept_year_n1992
concept_country_united_states , concept:atdate , concept_year_n1997
concept_country_united_states , concept:organizationhiredperson , concept_personmexico_ryan_whitney
concept_city_home , concept:atdate , concept_year_n1992
concept_city_home , concept:atdate , concept_year_n1997
concept_city_home , concept:organizationhiredperson , concept_personmexico_ryan_whitney
concept_city_service , concept:atdate , concept_year_n1992
concept_city_service , concept:atdate , concept_year_n1997
concept_city_service , concept:organizationhiredperson , concept_personmexico_ryan_whitney
concept_country_left_parties , concept:worksfor_inv , concept_personmexico_ryan_whitney
concept_country_left_parties , concept:organizationhiredperson , concept_personmexico_ryan_whitney
concept_year_n1992 , concept:atdate_inv , concept_governmentorganization_house
concept_year_n1992 , concept:atdate_inv , concept_country_united_states
concept_year_n1992 , concept:atdate_inv , concept_city_home
concept_year_n1992 , concept:atdate_inv , concept_tradeunion_congress
concept_year_n1997 , concept:atdate_inv , concept_governmentorganization_house
concept_year_n1997 , concept:atdate_inv , concept_country_united_states
concept_year_n1997 , concept:atdate_inv , concept_city_home
concept_personmexico_ryan_whitney , concept:worksfor , concept_governmentorganization_house
concept_personmexico_ryan_whitney , concept:worksfor , concept_tradeunion_congress
concept_personmexico_ryan_whitney , concept:worksfor , concept_country_left_parties
concept_governmentorganization_house , concept:personbelongstoorganization_inv , concept_personus_party
concept_governmentorganization_house , concept:worksfor_inv , concept_personmexico_ryan_whitney
concept_governmentorganization_house , concept:organizationhiredperson , concept_personmexico_ryan_whitney
concept_tradeunion_congress , concept:organizationhiredperson , concept_personus_party
```

```
concept_tradeunion_congress , concept:worksfor_inv , concept_personmexico_ryan_whitney
concept_tradeunion_congress , concept:organizationhiredperson , concept_personmexico_ryan_whitney
concept_country_left_parties , concept:organizationhiredperson , concept_personus_party
```

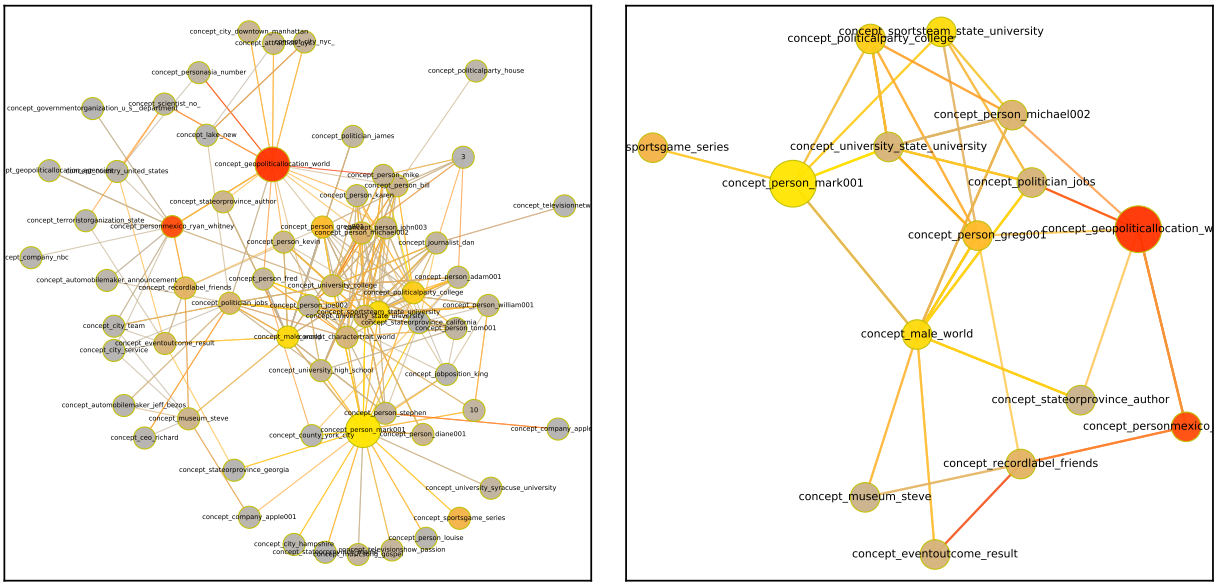


Figure 19: **AgentBelongsToOrganization**. The head is *concept_person_mark001*, the query relation is *concept:agentbelongstoorganization*, and the tail is *concept_geopoliticallocation_world*. The left is a full subgraph derived with *max_attending_from_per_step*=20, and the right is a further pruned subgraph from the left based on attention. The big yellow node represents the head, and the big red node represents the tail. Color on the rest indicates attention scores over a T-step reasoning process, where grey means less attention, yellow means more attention gained during early steps, and red means gaining more attention when getting closer to the final step.

For the AgentBelongsToOrganization task

```
Query: (concept_person_mark001 , concept:agentbelongstoorganization , concept_geopoliticallocation_world)

Selected key edges:
concept_person_mark001 , concept:personbelongstoorganization , concept_sportsteam_state_university
concept_person_mark001 , concept:agentcollaborateswithagent , concept_male_world
concept_person_mark001 , concept:agentcollaborateswithagent_inv , concept_male_world
concept_person_mark001 , concept:personbelongstoorganization , concept_politicalparty_college
concept_sportsteam_state_university , concept:personbelongstoorganization_inv , concept_politician_jobs
concept_sportsteam_state_university , concept:personbelongstoorganization_inv , concept_person_mark001
concept_sportsteam_state_university , concept:personbelongstoorganization_inv , concept_person_greg001
concept_sportsteam_state_university , concept:personbelongstoorganization_inv , concept_person_michael002
concept_male_world , concept:agentcollaborateswithagent , concept_politician_jobs
concept_male_world , concept:agentcollaborateswithagent_inv , concept_politician_jobs
concept_male_world , concept:agentcollaborateswithagent , concept_person_mark001
concept_male_world , concept:agentcollaborateswithagent_inv , concept_person_mark001
concept_male_world , concept:agentcollaborateswithagent , concept_person_greg001
concept_male_world , concept:agentcollaborateswithagent_inv , concept_person_greg001
concept_male_world , concept:agentcontrols , concept_person_greg001
concept_male_world , concept:agentcollaborateswithagent , concept_person_michael002
concept_male_world , concept:agentcollaborateswithagent_inv , concept_person_michael002
concept_politicalparty_college , concept:personbelongstoorganization_inv , concept_person_mark001
concept_politicalparty_college , concept:personbelongstoorganization_inv , concept_person_greg001
concept_politicalparty_college , concept:personbelongstoorganization_inv , concept_person_michael002
concept_politician_jobs , concept:personbelongstoorganization , concept_sportsteam_state_university
concept_politician_jobs , concept:agentcollaborateswithagent , concept_male_world
concept_politician_jobs , concept:agentcollaborateswithagent_inv , concept_male_world
concept_politician_jobs , concept:worksfor , concept_geopoliticallocation_world
concept_person_greg001 , concept:personbelongstoorganization , concept_sportsteam_state_university
concept_person_greg001 , concept:agentcollaborateswithagent , concept_male_world
concept_person_greg001 , concept:agentcollaborateswithagent_inv , concept_male_world
concept_person_greg001 , concept:agentcontrols_inv , concept_male_world
concept_person_greg001 , concept:agentbelongstoorganization , concept_geopoliticallocation_world
```

concept_person_greg001 , concept:personbelongstoorganization , concept_politicalparty_college
concept_person_greg001 , concept:agentbelongstoorganization , concept_recordlabel_friends
concept_person_michael002 , concept:personbelongstoorganization , concept_sportsteam_state_university
concept_person_michael002 , concept:agentcollaborateswithagent , concept_male_world
concept_person_michael002 , concept:agentcollaborateswithagent_inv , concept_male_world
concept_person_michael002 , concept:agentbelongstoorganization , concept_geopoliticallocation_world
concept_person_michael002 , concept:personbelongstoorganization , concept_politicalparty_college
concept_geopoliticallocation_world , concept:worksfor_inv , concept_personmexico_ryan_whitney
concept_geopoliticallocation_world , concept:organizationhiredperson , concept_personmexico_ryan_whitney
concept_geopoliticallocation_world , concept:worksfor_inv , concept_politician_jobs
concept_recordlabel_friends , concept:organizationhiredperson , concept_personmexico_ryan_whitney
concept_personmexico_ryan_whitney , concept:worksfor , concept_geopoliticallocation_world
concept_personmexico_ryan_whitney , concept:organizationhiredperson_inv , concept_geopoliticallocation_world
concept_personmexico_ryan_whitney , concept:organizationhiredperson_inv , concept_recordlabel_friends

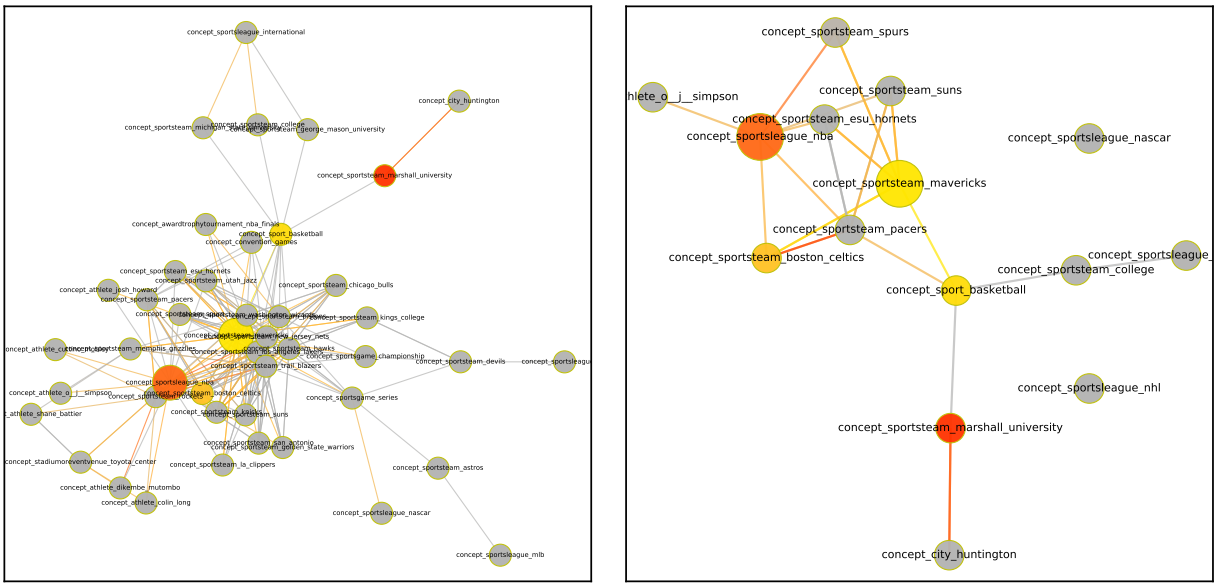


Figure 20: **TeamPlaysInLeague**. The head is *concept_sportsteam_mavericks*, the query relation is *concept:teamplaysinleague*, and the tail is *concept_sportsleague_nba*. The left is a full subgraph derived with *max_attending_from_per_step*=20, and the right is a further pruned subgraph from the left based on attention. The big yellow node represents the head, and the big red node represents the tail. Color on the rest indicates attention scores over a T-step reasoning process, where grey means less attention, yellow means more attention gained during early steps, and red means gaining more attention when getting closer to the final step.

For the TeamPlaysInLeague task

Query: (concept_sportsteam_mavericks , concept:teamplaysinleague , concept_sportsleague_nba)

Selected key edges:
concept_sportsteam_mavericks , concept:teamplayssport , concept_sport_basketball
concept_sportsteam_mavericks , concept:teamplaysagainstteam , concept_sportsteam_boston_celtics
concept_sportsteam_mavericks , concept:teamplaysagainstteam_inv , concept_sportsteam_boston_celtics
concept_sportsteam_mavericks , concept:teamplaysagainstteam , concept_sportsteam_spurs
concept_sportsteam_mavericks , concept:teamplaysagainstteam_inv , concept_sportsteam_spurs
concept_sport_basketball , concept:teamplayssport_inv , concept_sportsteam_college
concept_sport_basketball , concept:teamplayssport_inv , concept_sportsteam_marshall_university
concept_sportsteam_boston_celtics , concept:teamplaysinleague , concept_sportsleague_nba
concept_sportsteam_spurs , concept:teamplaysinleague , concept_sportsleague_nba
concept_sportsleague_nba , concept:agentcompeteswithagent , concept_sportsleague_nba
concept_sportsleague_nba , concept:agentcompeteswithagent_inv , concept_sportsleague_nba
concept_sportsteam_college , concept:teamplaysinleague , concept_sportsleague_international