Aligning Path-based Link Prediction with Human Understanding of Valid Reasoning

Anonymous authors
Paper under double-blind review

Abstract

Path-based link prediction methods reconstruct missing links between two vertices of a knowledge graph. They reconstruct a missing link by finding a path through the knowledge graph connecting both vertices. The path is the reasoning of the link prediction method. However, path-based link prediction methods are vulnerable to *Clever Hans* biases. They learn invalid reasoning patterns if these patterns are dominant and generalize well to the training and validation set. As a result, performance drops when evaluated on the real-world distribution. The validity of reasoning is determined by the semantic concept underlying the missing link, which is mostly accessible through human knowledge. The paper's approach makes human understanding of valid reasoning accessible while learning to predict missing links. This paper proposes the path-based link prediction method *LiEr*. *LiEr* learns valid reasoning within the knowledge graph domain from preference-based human feedback. The paper demonstrates that *LiEr*'s prediction capability is on par with other state-of-the-art link prediction methods while more aligned with human understand

ing of valid reasoning on various benchmark reasoning tasks. In addition, a novel benchmark knowledge graph with a *Clever Hans* bias is introduced to evaluate the alignment of link prediction methods with human understanding of valid reasoning. The paper contributes by brining human-in-the-loop learning to link prediction, and proposing a method capable of aligning its reasoning with the human understanding of valid reasoning.

1 Introduction

Path-based link prediction methods learn paths within a knowledge graph that robustly reconstruct missing links between vertices (Das et al., 2018; Wan et al., 2020). The paths are the explanations for the link prediction, as they are a human-interpretable mapping of the input to the output (Bhowmik & de Melo, 2020; Bahr et al., 2025b; Wehner et al., 2023). For example, to predict that `Berlin` is in `Germany`, a path such as `Berlin - locatedInState` → `Brandenburg - partOfCountry` → `Germany` provides a clear, human-interpretable sequence of relations which maps the input entities to the predicted link. The paths are the observable reasoning of the link prediction (Lin et al., 2018). The user of a path-based link prediction method expect the reasoning to be valid (Copi, 1954). **Valid reasoning** means that the truth of each reasoning step makes it impossible for the conclusion to be false (Copi, 1954). In other words, each reasoning step is relevant to the conclusion. The knowledge graph is a factual system (Hogan et al., 2022). Thus, all reasoning steps of a path-based link prediction method are true. However, the composition of all steps may constitute **invalid reasoning**: even if each step is individually true, some may be irrelevant to the conclusion because of spurious relations (i.e. edges that are factually correct but causally or semantically irrelevant to the prediction) in the graph (Copi, 1954).

Consider the following example (cf. Figure 1). A path-based link prediction method predicts `Hungary` to be in `Europe` because `Hungary` consumes pepper from `Europe`. The conclusion that `Hungary` is in `Europe` is true. Also, the reasoning step that `Hungary` consumes pepper from `Europe` is true. However, `Hungary` consuming pepper from `Europe` does not imply whether `Hungary` is part of `Europe` or not. The reasoning is invalid. A preferable reasoning pattern is that `Hungary` is in the region of `Central Europe`, and `Central Europe` is in `Europe`. Here, the conclusion is true, and the reasoning is valid as it is relevant to the conclusion.

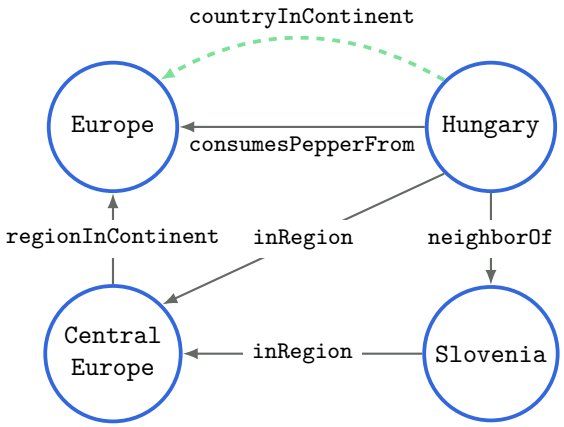

Figure 1: The figure depicts an excerpt from a knowledge graph about countries, regions, and continents. A path-based link prediction method predicts the missing link `countryInContinent` (green) from `Hungary` to `Europe` based on learned reasoning patterns connecting the two vertices.

Invalid reasoning patterns are likely to be learned by path-based link prediction methods if the patterns are frequent (dominant) and thus yield high accuracy (robust) on the train and validation set (Marconato et al., 2023), as will be demonstrated in Section 4. This is an alignment problem rooted in the discrepancy between what users expect the model to learn and what it actually learns (Christian, 2020). The users of the link prediction method expect it to learn a reasoning that reflects their knowledge, and thus understanding, of the semantic concept underlying the missing link (e.g., what it means for a country to be in a continent). However, the semantic concept is not explicitly included in the supervision signal given to the link prediction method while training, so it is not directly enforced during learning. Instead, invalid reasoning patterns, semantically unrelated to the classification goal, are potentially being learned if they frequently reconstruct a missing link on the training and validation sets. Meanwhile, these invalid reasoning patterns do not hold outside the training and validation splits; models that rely on them fail to generalize to real-world data. This is the **Clever Hans bias** (Lapuschkin et al., 2019): like the horse that seemed to do arithmetic by reading its trainer's cues, a link predictor can appear accurate by exploiting irrelevant, dataset-specific correlations instead of performing genuine, valid reasoning.

This paper proposes *LiEr*[1] (**L**earning **i**nteractively from **E**xplanations to **r**eason). *LiEr* is a path-based link prediction method that learns iteratively and interactively to reason (cf. Figure 2). It is optimized to reflect the human understanding of valid reasoning within the knowledge graph domain. This is achieved by learning a reward function iteratively from preference-based human feedback. A policy network proposes reasoning paths that maximize the expected reward.

First, the paper introduces the formal notion of knowledge graphs and of the link prediction task. Next, it describes how the link prediction task is reformulated to a *Markov Decision Process (MDP)* (Bellman, 1957). The paper presents a non-stationary history-dependent policy to solve the *MDP*. It follows up by introducing the formal properties of the reward function, which approximate the human understanding of valid reasoning. It describes how preference-based feedback over reasoning-pairs is collected from a human and how it is used to optimize the reward function towards rewarding valid reasoning. In addition, we argue why preference-based feedback is particularly suitable to approximate human understanding of valid reasoning in path-based link prediction. Furthermore, the paper describes the training of the policy network. *LiEr* is evaluated on several knowledge graphs from reasoning-heavy domains to demonstrate that its reasoning capabilities are on-par with state-of-the-art link prediction methods. In addition, the paper introduces a novel benchmark knowledge graph, called *Clever Hans' Countries*. *Clever Hans' Countries* is designed to illustrate that *LiEr* aligns its reasoning with human understanding of valid reasoning, leading to a performance boost in knowledge graphs with a high potential for link prediction models to learn *Clever Hans* biases (Lapuschkin

[1]It is commonly believed that Li Er is the birth name of the semi-legendary philosopher Laozi, the founder of Taoism (Seidel & von Falkenhausen, 2008).

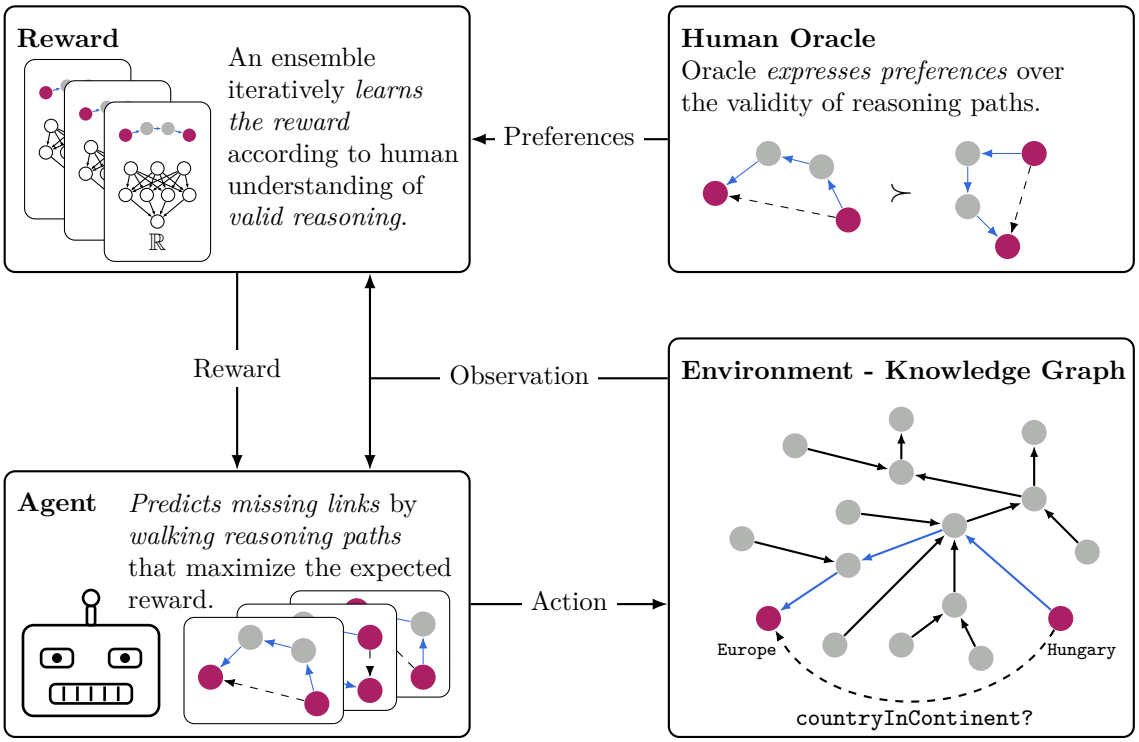

Figure 2: The schematic overview of *LiEr*. *LiEr* is a reinforcement learning agent that walks through a knowledge graph starting at a given head entity, finding the most plausible tail entity for a missing relation. The walk resembles a reasoning process aligned with human knowledge of the semantic concept underlying the missing link. *LiEr* realizes this alignment with the help of an ensemble of reward estimators trained on preferences from a human oracle over the reasoning paths.

et al., 2019). Finally, the paper provides observations and practical implications from preference-based feedback collection. The paper contributes by:

- Proposing, to the best of our knowledge, the first preference-based human-in-the-loop method for path-based link prediction in knowledge graphs, based on preference-based reinforcement learning from human feedback (Christiano et al., 2017; Ouyang et al., 2022).
- Demonstrating how to align the reasoning of a link prediction with human understanding of valid reasoning.
- Describing the *Clever Hans* bias in knowledge graph completion and proposing a benchmark knowledge graph that captures the bias in a controlled manner.

2 Related Work

Research on path-based link prediction and human-in-the-loop learning inspired *LiEr*. Link prediction is organized in embeddings-based methods, symbolic logic-based methods, and neuro-symbolic link prediction methods (Zhang et al., 2021). Embedding-based methods, such as *TransE* (Bordes et al., 2013), *DistMult* (Yang et al., 2015), *ConvE* (Dettmers et al., 2018), and *RoEMF* (Lu et al., 2025) demonstrate good performance while scaling to large knowledge graphs. However, by embedding the knowledge graph into sub-symbolic space, embedding-based methods lose the expressive topology of the knowledge graph, resulting in uninterpretable predictions that do not necessarily adhere to principles of reasoning (Schramm et al., 2023; Schramm & Schmid, 2023; Wehner et al., 2025).

Symbolic reasoning and logic-based link prediction methods such as *AIME+* (Galárraga et al., 2015), *ScaLeKB* (Chen et al., 2016), *SAFRAN* (Ott et al., 2021), and *AnyBURL* (Meilicke et al., 2024) learn explicit rules to predict missing links via reasoning. They operate entirely in the symbolic space, using the expressive topology of a knowledge graph. However, symbolic reasoning methods fall short if nuanced link prediction behavior for individual vertices is required, as they are limited in capturing special features of individual vertices (Schramm et al., 2023).

Neuro-symbolic link prediction methods, such as *NTP-λ* (Rocktäschel & Riedel, 2017), *NeuralLP* (Yang et al., 2017), *DRUM* (Sadeghian et al., 2019), and *NCRL* (Cheng et al., 2023) integrate symbolic and sub-symbolic methods to leverage the interpretability and structure of explicit rules while also benefiting from the scalability and statistical insights of sub-symbolic approaches. One particular line of research is path-based methods, such as *MINERVA* (Das et al., 2018), *Multi-Hop* (Lin et al., 2018), and *CURL* (Zhang et al., 2022). They reformulate the knowledge graph as a *MDP* (Bellman, 1957) to learn reasoning paths (Wan et al., 2020) that reconstruct plausible missing links, optimized through reinforcement learning.

Embedding-based, symbolic, and neuro-symbolic link prediction methods learn to predict missing links from patterns in the knowledge graph, optimizing for performance on the validation set, which serves as their only ground truth (Marconato et al., 2023). Thus, link prediction methods are vulnerable to learning invalid reasoning that does not generalize well to the real-world distribution (cf. Section 4.4), if invalid reasoning patterns consistently perform well on the training and validation set (Marconato et al., 2023). This is also called the *Clever Hans* bias (Lapuschkin et al., 2019) and can be understood as an alignment problem. In particular, there is a misalignment in the model's high performance on standard evaluation metrics while not learning valid reasoning patterns, which is what users actually expect from the model.

Artificial intelligence (*AI*) alignment research roots human intended goals and preferences into *AI* systems (Wiener, 1960; Christian, 2020; Gabriel, 2020; Teso et al., 2023; Ilievski et al., 2025). Developers and users of link prediction methods expect valid reasoning. However, the validity of reasoning depends on human knowledge of the semantic concepts underlying the missing link. The semantic concept is not explicitly part of the knowledge graph and, thus, can only be approximated. The approximation is realized by reconstructing tail mappings. The consequence is a misalignment between the *AI* system's intended goal, valid reasoning, and its realization via tail mapping reconstruction. This leads to unintended behavior of the link prediction method if the knowledge graph is exploitable with a *Clever Hans* bias (Lapuschkin et al., 2019). A large corpus of literature on incorporating human intent in machine-learning methods exists (Mosqueira-Rey et al., 2023; Najar & Chetouani, 2021). In particular, human-in-the-loop reinforcement learning approaches like *TAMER+RL* (Knox & Stone, 2011), deep reinforcement learning from human preferences (Christiano et al., 2017), *EXPAND* (Guan et al., 2020), and *MAPLE* (Mahmud et al., 2024) make significant progress in aligning the behavior of reinforcement agents in physical domains with human intent. Preference-based human-in-the-loop learning also showed successful applications in *natural language processing* with *InstructGPT* (Ouyang et al., 2022). This paper aims to use preference-based human-in-the-loop learning to align the reasoning of a link prediction with human understanding of valid reasoning.

3 Aligning Link Prediction with Human Understanding of Valid Reasoning

In this paper, we introduce *LiEr*, a path-based link prediction method that is trained to align its internal reasoning with human understanding of valid reasoning. Section 3.1 formalizes the knowledge graph and defines the missing link prediction task of *LiEr*. This sets the foundation for the detailed description of *LiEr* in Section 3.2.1. In Section 3.2.1, the knowledge graph is translated into a *MDP*. Section 3.2.2 explains the policy learning method used to solve the *MDP*, given a reward function ensuring alignment with human reasoning (cf. Section 3.2.3). This section aims to fully describe the method behind *LiEr*'s preference-based reward alignment.

3.1 Defining the Knowledge Graph and Link Prediction Task for LiEr

The setting of *LiEr* is a knowledge graph. The knowledge graph $\mathcal{G}$ is defined as a directed, labeled, multigraph (Hogan et al., 2022; Das et al., 2018). Thus, the knowledge graph is a typed quiver (Assem et al.,

2006)

$$\mathcal{G} = (V, E, \Sigma_V, \Sigma_E, h, t, \ell_V, \ell_E), \tag{1}$$

with a set V of vertices $v \in V$ (i.e., entities/nodes), a set E of edges $e \in E$ (i.e., relations/links), an alphabet Σ_V with symbols for the vertices $\sigma_v \in \Sigma_V$, an alphabet Σ_E for the edge symbols $\sigma_e \in \Sigma_E$. The head mapping $h : E \to V$ maps an edge e to its head vertex v_h, and the tail mapping $t : E \to V$ maps an edge e to its tail vertex v_t. The tail and head mapping enforce the directionality of the graph. The vertex language mapping $\ell_V : V \to \Sigma_V$ maps a vertex v to its corresponding symbol σ_v, and the edge language mapping $\ell_E : E \to \Sigma_E$ maps an edge e to its corresponding symbol σ_e (Gallian, 2000; Harary et al., 1965).

To illustrate this thorough definition, consider a simple example knowledge graph with three vertices $V = \{v_1, v_2, v_3\}$. To assign meaningful names to these vertices, we define the vertex alphabet $\Sigma_V = \{\texttt{Poland}, \texttt{Central Europe}, \texttt{Europe}\}$. The vertex language mapping ℓ_V associates each vertex with its corresponding name. For example, $\ell_V(v_1)$ returns `Poland`.

Next, we define the edges that represent relations between vertices $E = \{e_1, e_2\}$. Their labels are provided by the edge alphabet $\Sigma_E = \{\texttt{neighbour}, \texttt{locatedIn}\}$. The edge language mapping ℓ_E associates each edge with its corresponding label. For instance, $\ell_E(e_1)$ returns `locatedIn`.

Finally, the connectivity between vertices is established by the head and tail mappings. The head mapping h returns the starting vertex of an edge, while the tail mapping t returns its ending vertex. Suppose that $h(e_1) = v_1$ and $t(e_1) = v_2$. With the help of this definition, we establish a structured way to retrieve the fact that `Poland is located in Central Europe`. Additionally, this structured approach enables the subsequent translation of the knowledge graph into a traversable *MDP*.

The goal of LiEr is to predict missing links. *LiEr* learns a mapping

$$p : V \times \Sigma_E \to V, \tag{2}$$

that predicts for any given query q, consisting of a head vertex v_h^q and edge symbol σ_e^q, the most plausible tail vertex v_t^q, referred to as the *answer*. The query q is frequently represented as a triple $(v_h^q, \sigma_e^q, ?)$. The question mark in q denotes the return value of p, which is the most plausible tail vertex.

The link prediction mapping p is selected from the set of all possible mappings $p \in P$ to approximate the tail mapping t as accurately as possible on a validation edge set $E_{valid} \subset E$. Satisfying Equation 3 objectivizes the link prediction mapping p towards a consistently high performance over unseen queries (i.e., validation data). Thus, the link prediction mapping p aims to learn reasoning patterns that generalize well to all possible and correct query-answer pairs

$$\arg\max_{p \in P} (|\{e | \forall e \in E_{valid} : t(e) = p(h(e), \ell_E(e))\}|). \tag{3}$$

The advantage of the tail link prediction mapping p over tail mapping t is that p does not require an edge between v_h^q and v_t^q to determine v_t^q as the correct tail vertex. The link prediction mapping p solely relies on a head vertex v_h^q and an edge symbol σ_e^q. This allows p to return v_t^q even if an edge e with the symbol σ_e^q between v_h^q and v_t^q is missing.

Reconsider the simple knowledge graph from the previous example. Suppose that we want to determine the continent in which `Poland` is located. In the current graph, the corresponding edge (i.e., the relation `locatedIn`) linking `Poland` to its continent is missing. Thus, the tail mapping cannot be used to retrieve an answer.

To address this gap, we introduce the link prediction mapping p. The mapping takes a query in the following manner: $q = (\texttt{Poland}, \texttt{locatedIn}, ?)$, as input. The link prediction mapping p processes the query q based on its learned patterns and returns the most plausible vertex candidate for the missing link, ideally returning `Europe` as the answer.

As described in the introduction, this paper proposes an approach that allows for human-in-the-loop learning of the link prediction mapping p. The aim is to iteratively improve the reasoning employed by p to arrive at an answer v_t^q with the help of human feedback.

3.2 Predicting Missing Links with LiEr by Learning Valid Reasoning from Human Feedback

The following formulates the link prediction task as an *MDP* (Bellman, 1957), similar to previous research on path-based link prediction methods (Das et al., 2018; Lin et al., 2018; Wan et al., 2020). A knowledge graph can easily be understood as an MDP, by looking at its nodes as states and its edges as actions. This enables the optimization of the link prediction mapping p according to Equation 3 via reinforcement learning. For that reason, Section 3.2.1 describes how the knowledge graph is transformed into an *MDP*. Section 3.2.2 details the policy learning strategy used to solve the *MDP*, which leverages a reward function specifically designed to align with human reasoning, explained in Section 3.2.3. Finally, the training procedure for *LiEr* is outlined, describing the optimization of its preference-based reward.

3.2.1 From Knowledge Graph to Markov Decision Process

The *Markov Decision Process* is defined as a 4-tuple $MDP = (\mathcal{S}, \mathcal{A}, \delta, \mathcal{R})$.

The *MDP* holds a state space $\mathcal{S} \subseteq \{V \times V \times \Sigma_E\}$. A state $s_c \in \mathcal{S}$ at step c is defined as $s_c = (v_c, v_h^q, \sigma_e^q)$. It consists of the vertex v_c and (v_h^q, σ_e^q), which is the input to p (cf. Equation 2).

A set of actions $\mathcal{A}_{c_s} \subseteq \mathcal{A}$ is available at each state s_c.

$$\mathcal{A}_{c_s} = \{(v_c, e, v) | \forall e \in E, \forall v \in V : h(e) = v_c \wedge t(e) = v\} \cup \{(v_c, e_{loop}, v_c)\} \tag{4}$$

The 3-tuple (v_c, e, v) is called an action a. Intuitively, an agent can *walk* any of the outgoing edges e from vertex v_c to arrive at e's tail node v. In addition, a loop action (v_c, e_{loop}, v_c) is available. The loop action allows staying at a vertex v_c. This is useful in cases where staying at v_c maximizes the expected reward.

The mapping $\delta : \mathcal{S} \times \mathcal{A} \to \mathcal{S}$ defines the transition from the current state s_c to the follow-up state s_{c+1}, given action $a_c = (v_c, e, v_{c+1})$:

$$\delta(s_c, a_c) = (v_{c+1}, v_h^q, \sigma_e^q) = s_{c+1}. \tag{5}$$

The tail vertex v_{c+1} of the selected action a_c becomes the current vertex of the following step $c + 1$. The transition probability $P(s_{c+1}|s_c, a_c) = 1$ is fully deterministic.

Finally, the *MDP* includes a reward function $\mathcal{R} : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$. The reward $\mathcal{R}$ takes as input a state $s \in \mathcal{S}$, and an action $a \in \mathcal{A}$ and returns a numeric reward. Section 3.2.3 describes how $\mathcal{R}$ is parameterized and trained to be aligned with human understanding of valid reasoning.

Once the knowledge graph is transformed into an *MDP*, a policy has to be defined that describes and governs the walk through the *MDP* to optimize the link prediction mapping defined in Equation 3 that returns the most plausible answers to a link prediction query. This process is called solving the *MDP*.

The reframing of the link prediction task as a *MDP* is now fully described. The following section introduces how the *MDP* is solved to predict missing links.

3.2.2 Solving the Markov Decision Process

The following describes the solving process of the *MDP*. To get an explicit solution, a non-stationary history-dependent policy $\pi = (d_1, \ldots, d_c)$ is implemented. To solve the policy π, at each step c, a policy network calculates a probability distribution d_c over all available actions $\mathcal{A}_{c_s}$ defined by:

$$h_c^d = \text{lnLSTM}(h_{c-1}^d, [\alpha_{c-1}; \varsigma_c]) \tag{6}$$

$$d_c = \text{Softmax}(A_c \cdot \text{leakyReLU}(W_2 \cdot \text{leakyReLU}(W_1 \cdot [h_c^d; \varsigma_c] + B_1) + B_2)). \tag{7}$$

The recurrent neural network architecture in Equation 6 and 7 describes the policy network. The recurrent architecture enables the policy network to consider previously encountered states and actions while choosing the next action (cf. Figure 3). At step c, a layer normalized long short-term memory ($lnLSTM$) (Ba et al., 2016) calculates the history $h_c^d \in \mathbb{R}^m$ (cf. Equation 6). The $lnLSTM$ takes two inputs. The first input is the previous history h_{c-1}^d. The second input is a stacking of the previously executed actions' (a_{c-1}) embedding $\alpha_{c-1} \in \mathbb{R}^n$ and the current states' (s_c) embedding $\varsigma_c \in \mathbb{R}^l$. The numbers $m, n, l \in \mathbb{N}$

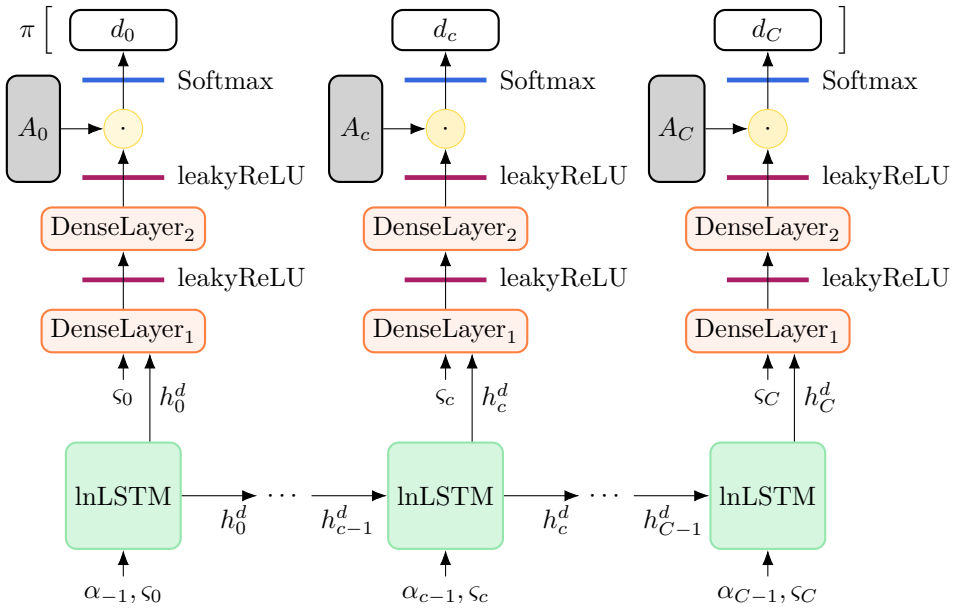

Figure 3: The policy network architecture of *LiEr* operates as follows. At each step c, with $0 \leq c \leq C$, the probability distribution d_c, as part of the policy π, is calculated using a layer-normalized long-short term memory ($lnLSTM$). This is followed by two dense layers that utilize leaky ReLU activations. The output from these layers is then multiplied ($\cdot$) by the stacked embeddings of all available actions A_c. Finally, this results in the probability distribution d_c through a softmax function (cf. Glossary 10).

are the lengths of the embeddings. Next, the policy network stacks h_c^d with ς_c. The stacked embedding is first passed to a dense layer (Goodfellow et al., 2016) with weights W, biases B, and a leaky rectified linear ($leakyReLU$) activation (Redmon et al., 2016). This is followed by another dense layer that also uses a $leakyReLU$ activation. The embeddings of all available actions $\mathcal{A}_{c_s}$ are stacked in $A_c \in \mathbb{R}^{|\mathcal{A}_{c_s}| \times n}$ and multiplied with the output of the previous dense layer. This results via a *softmax* function (Goodfellow et al., 2016) in a probability distribution $d_c \in \mathbb{R}^{|\mathcal{A}_{c_s}|}$ over all available actions $\mathcal{A}_{c_s}$ (cf. Equation 7).

To calculate the policies for every step, the agent starts at the head vertex of the query v_h^q and chooses an action $a_c \in \mathcal{A}_{c_s}$ according to the probability distribution d_c. In cases of exploration, this means that an action is sampled from all available actions according to the probability distribution. This is the dominant strategy while training. In exploitation cases, such as the evaluation, the action with the highest probability is chosen. This is the fuzzy reasoning step of *LiEr*. The transition is applied as described in Equation 5, and the process is repeated at the next state with its available actions and an updated history. The agent stops after a finite and predefined number of repetitions C (i.e., steps), with $0 \leq c < C$. The final vertex in the final state $v_C \in s_C$ after all steps are executed is the return vertex v_t^q of the link prediction mapping p (cf. Equation 3).

Let the composition of all reasoning steps $\kappa_c = (s_c, a_{c-1})$ be a path $w = (\kappa_1, ..., \kappa_c, ..., \kappa_C)$. The path reflects the reasoning applied by the link prediction mapping p to arrive at the answer. It maps the query to the answer. Thus, the path is the policy applied to solve the *MDP* and the explanation of the link prediction.

On aligning the reasoning steps with human understanding of valid reasoning. Notably, A_c binds the sub-symbolic policy network to the topology of the knowledge graph, a syntactically correct and symbolic system. This results in individual steps of the agent being true and interpretable. Meanwhile, the composition of all steps may still lead to a wrong query answer. In particular, the composition of all steps may not reflect human understanding of valid reasoning regarding their knowledge about the semantic concept underlying the missing link.

The policy network's parameters are trained to maximize the expected reward. We use this to align *LiEr*s' reasoning paths with human understanding of valid reasoning.

The next section explains how the reward $\mathcal{R}$ is modeled to guide the policy network toward reasoning that matches human understanding of the query's underlying concept.

3.2.3 Aligning the Policy Network with Human Understanding of Valid Reasoning via the Reward

This section proposes the alignment of the policy network with valid reasoning, by the reward $\mathcal{R}$.

Domain experts generally know valid reasoning within their domain. However, they may not know how domain knowledge is formalized within the knowledge graph. For example, the domain expert may know that `Hungary` is in `Europe` because `Hungary` is in `Central Europe` and `Central Europe` is in `Europe`. However, they may not know that "`Hungary` is in `Central Europe`" is formalized via the `locatedIn` relationship (i.e., `locatedIn(Hungary, Central Europe)`). This makes it difficult for domain experts to give rules or examples according to the formalization enforced by the knowledge graph. In addition, some domain knowledge is tacit. Valid reasoning patterns may be difficult for the expert to verbalize. However, if the domain experts see examples, they can judge the correctness and validity.

For that reason, we propose a preference-based feedback approach that automatically creates reasoning examples for the domain expert to judge. This leaves it to the domain expert to establish a ranking over the reasoning examples, going from valid and correct to incorrect examples. However, asking domain experts to rank many reasonings by categories or numeric values is cumbersome. Thus, the paper's approach uses pairwise preference collection, making it convenient for the expert to create a ranking. The preference-based approach enables a user-friendly collection of preferences and thus provides an efficient way to learn the human oracle's Λ understanding of valid reasoning.

Central to the pairwise preference alignment approach is the reward $\mathcal{R}$. The rewards are computed using an ensemble of recurrent neural networks (Dietterich et al., 2002), each computing r_c^i for the individual reasoning step κ_c. The computation is carried out as follows

$$h_c^r = \text{lnLSTM}\left(h_{c-1}^r, \varkappa_c\right), \tag{8}$$

$$r_c^i = \text{batchNorm}\left(\text{leakyReLU}\left(W_2 \cdot \text{leakyReLU}\left(W_1 \cdot [h_{c-1}^r; \varkappa_c] + B_1\right) + B_2\right)\right), \tag{9}$$

where $\varkappa \in \mathbb{R}^{l+n} = [\varsigma; \alpha]$ is the vector representation of κ, and l is the length of the state embedding ς, n the length of the action embedding α. The reward is normalized over the batch ($batchNorm$) to stabilize the training.

The training requires aligning the reward with the help of labels annotated by a human oracle. To make the alignment process as efficient as possible, we select samples of paths with the highest uncertainty in the reward for feedback. This ensures a large impact on the reward training. To calculate the uncertainty in the reward, a reward ensemble is used (Dietterich et al., 2002). The final reward $\hat{r_c}$ is the mean over the rewards r_c^i of each ensemble member i.

The reward ensemble is trained to enforce properties that approximate valid reasoning, based on a human's knowledge of the semantic concept underlying a missing link. The following paragraphs formally introduce those properties necessary for aligning *LiEr* with human understanding of valid reasoning.

Properties of $\hat{r}$. The reward $\hat{r}$ is optimized to incentivize choosing paths based on the preference provided by a human oracle Λ, following the approach outlined by (Christiano et al., 2017).

Let the reasoning paths $w_{\sigma_e}^1 = (\kappa_1^1, ..., \kappa_c^1, ..., \kappa_C^1)$ and $w_{\sigma_e}^2 = (\kappa_1^2, ..., \kappa_c^2, ..., \kappa_C^2)$ result from a maximum of two different queries q_1 and q_2 with the same edge symbol σ_e. The embedded reasoning steps $\varkappa^1$ and $\varkappa^2$ of $w_{\sigma_e}^1$ and $w_{\sigma_e}^2$ are stored in $\omega_{\sigma_e}^1 = (\varkappa_1^1, ..., \varkappa_c^1, ..., \varkappa_C^1)$ and $\omega_{\varsigma_e}^2 = (\varkappa_1^1, ..., \varkappa_c^1, ..., \varkappa_C^1)$.

Given two reasoning paths $w_{\sigma_e}^1$ and $w_{\sigma_e}^2$, $\hat{r}$ satisfies the following: The reasoning path $w_{\sigma_e}^1$ shall receive a higher reward $\hat{r}$ compared to $w_{\sigma_e}^2$, if the human oracle Λ favors $w_{\sigma_e}^1$ over $w_{\sigma_e}^2$, as detailed in Equation 10. Conversely, the path $w_{\sigma_e}^2$ will receive a higher reward, if Λ prefers $w_{\sigma_e}^2$ over $w_{\sigma_e}^1$, as indicated in Equation 11.

(*Ex.1*)	`inContinent(Hungary, Europe)` ← `inRegion(Hungary, Central Europe)` ∧ `inContinent(Central Europe, Europe)`
(*Ex.2*)	`inContinent(Hungary, Europe)` ← `consumesPepperFrom(Hungary, Europe)`
(*Ex.3*)	`inContinent(Hungary, Austria)` ← `hasNeighbor(Hungary, Austria)`

Table 1: The three exemplary reasonings *Ex.1*, *Ex.2*, and *Ex.3* are possible results for the query `(Hungary, inContinent, ?)`. The correct answer to the query is `Europe`. *Ex.1* and *Ex.2* arrive at this answer. However, *Ex.1*'s reasoning is valid and *Ex.2*'s is not.

If Λ is indifferent between the two reasoning paths, both $w^1_{\sigma_e}$ and $w^2_{\sigma_e}$ shall receive similar rewards, as stated in Equation 12.

$$\sum_{\forall \varkappa^1 \in \omega^1_{\sigma_e}} \hat{r}(\varkappa^1) > \sum_{\forall \varkappa^2 \in w^2_{\sigma_e}} \hat{r}(\varkappa^2) \text{ if } w^1_{\sigma_e} \succ_\Lambda w^2_{\sigma_e} \tag{10}$$

$$\sum_{\forall \varkappa^1 \in \omega^1_{\sigma_e}} \hat{r}(\varkappa^1) < \sum_{\forall \varkappa^2 \in w^2_{\sigma_e}} \hat{r}(\varkappa^2) \text{ if } w^1_{\sigma_e} \prec_\Lambda w^2_{\sigma_e} \tag{11}$$

$$\sum_{\forall \varkappa^1 \in \omega^1_{\sigma_e}} \hat{r}(\varkappa^1) \approx \sum_{\forall \varkappa^2 \in w^2_{\sigma_e}} \hat{r}(\varkappa^2) \text{ if } w^1_{\sigma_e} \sim_\Lambda w^2_{\sigma_e} \tag{12}$$

Properties of Λ. Two major factors lead the human oracle Λ while expressing preferences over paths:

1. *Correctness*: The answer is correct.
2. *Validity*: The truth of the reasoning steps is relevant to the truth of the conclusion (Copi, 1954).

Let us discuss the properties of Λ given the example from Table 1. In *Ex.1*, the reasoning is valid: Hungary is in Central Europe, and Central Europe is in Europe, so the conclusion that Hungary is in Europe is correct. In *Ex.2*, the conclusion is also correct, but the reasoning is invalid because it relies on Hungary consuming pepper from Europe, which is unrelated to the claim. Meanwhile, *Ex.3* concludes that Hungary is in Austria, which is incorrect. The human oracle prefers the correct and valid reasoning *Ex.1* over the correct but invalid reasoning *Ex.2* and reasoning *Ex.2* over the incorrect and invalid reasoning *Ex.3*. This results in the preference order $Ex.1 \succ Ex.2 \succ Ex.3$.

The policy network $\hat{r}$ is optimized to approximate the preference ordering of Λ by rewarding preferred reasoning higher compared to less preferred ones. Thus, $\hat{r}$ learns to reward reasoning steps by the *correctness* and *validity* of the resulting path.

Preference collection. The preference collection from a human oracle Λ requires pairs of reasoning paths. For that reason, pairs of paths $(w^1_{\sigma_e}, w^2_{\sigma_e})$ are sampled at each training epoch. The pairs are sampled such that the feedback required from Λ is minimized. This is done by selecting the $n \in \mathbb{N}$ pairs with maximum variance in the reward across all ensemble members (Christiano et al., 2017). Intuitively, pairs are selected and shown to Λ for feedback, for which the reward ensemble is most uncertain on how to reward them. This is done in the hopes of eradicating uncertainties and closing corresponding blind spots in the reward function. Finally, they are presented to the human oracle Λ (cf. Figure 4). Λ expresses preferences $(\succ \veebar \prec \veebar \sim)$ over the pairs in alignment with their understanding of *correctness* and *validity*. Every pair $(w^1_{\sigma_e}, w^2_{\sigma_e})$ and preference $(\succ_\Lambda \veebar \prec_\Lambda \veebar \sim_\Lambda)$ is stored in a database D for training the reward ensemble.

Fitting the Reward ensemble. At the core of the reward ensemble training is the loss function (cf. Equation 14). The loss function is similar to the loss proposed by (Christiano et al., 2017). It is based on the

Bradley-Terry model (Bradley & Terry, 1952). The *Bradley-Terry* model estimates a score function by observing pairwise preferences.

$$\hat{P}[w^1 \succ w^2] = \frac{\exp\left(\sum_{\forall \varkappa^1 \in \omega^1} \hat{r}(\varkappa^1)\right)}{\exp\left(\sum_{\forall \varkappa^1 \in \omega^1} \hat{r}(\varkappa^1)\right) + \exp\left(\sum_{\forall \varkappa^2 \in \omega^2} \hat{r}(\varkappa^2)\right)} \tag{13}$$

$$loss(\hat{r}) = -\sum_{\forall (w^1_{\sigma_e}, w^2_{\sigma_e}) \in D} (\mu * \log(\hat{P}[w^1_{\sigma_e} \succ w^2_{\sigma_e}]) + (1-\mu) * \log(\hat{P}[w^2_{\sigma_e} \succ w^1_{\sigma_e}])) \tag{14}$$

$$, \text{with } \mu = \begin{cases} 1.0 \text{ if } w^1_{\sigma_e} \succ_\Lambda w^2_{\sigma_e} \\ 0.0 \text{ if } w^1_{\sigma_e} \prec_\Lambda w^2_{\sigma_e} \\ 0.5 \text{ if } w^1_{\sigma_e} \sim_\Lambda w^2_{\sigma_e} \end{cases} \tag{15}$$

First, the probability of $\hat{r}$ preferring path w^1 over path w^2 and the probability of $\hat{r}$ preferring path w^2 over w^1 are computed (cf. Equation 13). Next, each probability is multiplied by the corresponding weighting factor. μ is used for the probability related to w^1, while $(1-\mu)$ is used for the probability related to w^2 (cf. Equation 14). The weighting factor μ is defined such that it equals 1 when Λ prefers w^1 over w^2, and it equals 0 when w^2 is preferred over w^1 (cf. Equation 15). If Λ is indifferent between the two paths, then μ is set to 0.5. This results in a loss that optimizes $\hat{r}$ to return a high reward if a path is preferred by Λ and a low reward if a path is not preferred by Λ. Thus, $\hat{r}$ is optimized to approximate Λ's preference ordering over all possible paths. The loss is calculated for all pairs in the database D and back-propagated through $\hat{r}$.

It is fully described how *LiEr* predicts links between two vertices via valid reasoning paths. The following section presents the experimental results.

4 Evaluation

The evaluation of *LiEr* is structured to test its ability to learn valid reasoning patterns from human feedback and to compare its predictive performance with established link prediction methods. To that end, *LiEr* is first benchmarked on the *Countries* knowledge graphs (Bouchard et al., 2015; Rocktäschel & Riedel, 2017) and its tasks. These are controlled environments, designed to test the reasoning capabilities of link prediction methods in varying degrees of difficulty. Similar to *MNIST* (Deng, 2012) in image classification, the *Countries* knowledge graph is narrowly scoped, highly structured, and easily understandable by humans. This makes it ideal for testing whether a method captures the basic mechanism it is designed for. In this case, we want to test *LiEr* on multi-step reasoning in knowledge graphs, without the confounding factors of scale, noise, or ambiguity. However, like *MNIST* or the *Cats vs. Dogs* (Parkhi et al., 2012) datasets, *Countries* is, due to its limited scope, not sufficient to assess real-world feasibility.

To assess performance in more complex and realistic settings, we extend the evaluation to three additional knowledge graphs: *family* (Yang et al., 2017), and the *NELL* (Mitchell et al., 2018) subsets *locations*, and *sports*. These datasets provide more heterogeneous structures and less constrained reasoning patterns, and serve to evaluate *LiEr*´s ability to generalize beyond synthetic benchmarks.

Finally, we are going to evaluate *LiEr*´s key claim: that preference-based feedback enables it to learn reasoning patterns aligned with human understanding of valid reasoning, even in the presence of spurious relations. To that end, the paper introduces the *Clever Hans Countries* knowledge graph, which contains intentionally constructed spurious correlations that are predictive on the training and validation data but fail to generalize to the test distribution.

In addition to predictive performance, we analyze the dynamics of *LiEr*´s interactive learning procedure. This includes measuring how much human feedback is required to train a well-performing model and how preference-based training behaves in the presence of label noise or inconsistent feedback.

4.1 Evaluation Setup

Implementation Details. *LiEr* is implemented in *PyTorch 1.10*. All evaluation were run on a workstation with an *AMD Ryzen Threadripper 2920X*, two *GeForce RTX 2080 TI*, and 64GB RAM. The policy network

(cf. Equations 6 and 7)) is trained using *REINFORCE* (Williams, 1992). The same adaptations as in Das et al. (2018) are used to calculate the expected reward and cost function. An additive control variate baseline reduces variance in the expected reward (Das et al., 2018; Evans et al., 2000; Hammersley, 2013). Furthermore, an entropy regularization term is added to the cost function to encourage a diverse sampling of reasoning paths during training time (Das et al., 2018). In addition, dropout is applied to all layer types and the available actions $\mathcal{A}_c^s$ to reduce overfitting (Goodfellow et al., 2016) and, again, to encourage a diverse sampling of reasoning paths during training time. The reward ensemble (cf. Equation 9) is trained with the *ADAM* optimizer (Kingma & Ba, 2015) and the Bradley-Terry loss (cf. Equation 14), an early stopping mechanism, and dropout is applied at every layer type to reduce overfitting (Goodfellow et al., 2016) [2].

Simulating the Oracle with Regions of Interest We simulate the oracle's preferences using predefined regions of interest (*ROIs*) defined per relation type in each knowledge graph to scale the feedback process under limited human availability. *ROI*s in knowledge graphs serve a role analogous to *ROI*s in image processing (Brinkmann, 2008): they define structured subregions that are relevant to the prediction task at hand. In our case, each *ROI* represents an abstract, valid reasoning pattern that plausibly reconstructs the missing link between head and tail entities via multi-hop paths. These regions are derived from domain knowledge and reflect reasoning patterns that align with human understanding of validity [3].

For example, in the *Countries* knowledge graph, a typical *ROI* for the relation `locatedIn` is the reasoning pattern:

$$\texttt{Country} \rightarrow \texttt{locatedIn} \rightarrow \texttt{Region} \rightarrow \texttt{locatedIn} \rightarrow \texttt{Continent}$$

Another *ROI* looks as follows:

$$\texttt{Country} \rightarrow \texttt{neighborOf} \rightarrow \texttt{Country} \rightarrow \texttt{locatedIn} \rightarrow \texttt{Continent}$$

Each *ROI* defines a reasoning template considered valid for the corresponding relation.

We generate preferences by assuming the oracle prefers reasoning paths that (i) follow an ROI over those that do not, and (ii) lead to the correct tail entity over those that do not. These assumptions let us derive preference labels automatically, that reflect correct and valid paths.

The *ROI*s are not a component of *LiEr*. *LiEr* consumes only pairwise preferences over reasoning paths and never accesses the *ROI*s directly. We use *ROI*-derived preferences to instantiate the human oracle so that our experiments are reproducible and do not require human annotators for every training run. Simulating the oracle with a known preference model is standard practice in preference-based reinforcement learning (Christiano et al., 2017; Lee et al., 2021); the assumption that the simulated oracle follows the *ROI*s is the analogue of the scripted-oracle assumption used there. As we show in Section 4.5 (Figure 5), *LiEr* does not require the simulated oracle to follow the *ROI*s perfectly. Performance degrades only gradually as the oracle's preferences deviate from the *ROI*s. This indicates that the method learns from the preference signal itself rather than from the *ROI* templates. All quantitative results reported in this paper (Tables 3–8 and Figure 5) are produced with *ROI*-derived preferences from the simulated oracle and do not include human-labelled preferences.

A more detailed description of how *ROI*s are defined for each knowledge graph is provided in the respective dataset sections.

Comparison Models. The predictive performance of *LiEr* is evaluated against several representative link prediction models that capture different methodological paradigms. As a neuro-symbolic baseline, we compare against *MINERVA* (Das et al., 2018), a non-interactive, path-based model that uses reinforcement

[2] A hyperparameter optimization study was conducted using *Optuna* (Akiba et al., 2019) for all hyperparameters, employing the *TPESampler* (Bergstra et al., 2011) to determine the best configurations for each dataset. For hyperparameter configurations specific to each knowledge graph, please refer to LiEr´s Github repository.

[3] The construction of *ROI*s is non-trivial and only feasible for relation types with well-defined semantics in structured domains. As a result, the creation of *ROI*s constrained the choice of benchmark knowledge graphs and directly influenced the selection of the seven knowledge graphs used in this work.

learning to discover multi-hop reasoning paths in knowledge graphs. In addition, we include *Neural Theorem Provers (NTP)* and its improved variant *NTP-λ* (Rocktäschel & Riedel, 2017), which perform differentiable logical inference via end-to-end training. We also consider *Neural Logic Programming (NeuralLP)* (Yang et al., 2017), which combines logic programming with differentiable rule learning. All three models provide interpretable reasoning chains and represent widely evaluated neuro-symbolic baselines for multi-hop reasoning. Their official implementations were used[4], with default hyperparameters unless dataset-specific configurations were available.

To contextualize *LiEr*'s performance among embedding-based models, we further compare against *TransE* (Bordes et al., 2013), *DistMult* (Yang et al., 2015), and *ConvE* (Dettmers et al., 2018). These models learn low-dimensional vector representations of entities and relations and are widely used due to their scalability and empirical performance. They do not model explicit reasoning chains but serve as baselines for general link prediction tasks. All embedding-based models were trained using the *PyKEEN* (Ali et al., 2021) framework[5], which offers standardized training pipelines and reproducible implementations of knowledge graph embedding methods.

The selection of comparison models aims to span the spectrum from reasoning to sub-symbolic embedding-based approaches, thereby providing a meaningful baseline for assessing *LiEr*'s reasoning capabilities, alignment behavior, and overall performance.

4.2 Evaluation Metrics

The following quantitative evaluation is based on two metrics: *Mean Reciprocal Rank (MRR)* and *Mean Reciprocal Localisation Rank (MRLR)*.

MRR is a standard metric in link prediction tasks that measures the average inverse rank of the first correct answer (Fuhr, 2018; Ali et al., 2021). For each query u, the model generates a ranked list of candidate entities, and the position of the first correct entity is recorded. A high MRR score (closer to 1) indicates that correct answers are consistently ranked near the top, reflecting strong predictive performance. Formally, the *MRR* is computed as:

$$\text{MRR} = \frac{1}{U} \sum_{u=1}^{U} \frac{1}{\text{rank}_u}$$

MRLR. To additionally assess the validity of reasoning paths, independent of whether the predicted answer is correct, we introduce the *Mean Reciprocal Localisation Rank (MRLR)*. The *MRLR* is structurally similar to the *MRR*, but instead of checking whether the predicted entity is correct, it evaluates whether the reasoning path leading to a prediction aligns with any of the predefined *ROI*s. For each ranked instance u, the corresponding reasoning path is extracted. If the path overlaps with any *ROI* associated with the query relation, the instance is considered valid. The rank of the highest-ranked valid reasoning path determines the contribution to the *MRLR*. Importantly, *MRLR* does not consider whether the predicted tail (or head) entity is correct. It is only concerned with the validity of the reasoning path. Consequently, *MRLR* can only be computed for models that produce explicit reasoning paths as part of their output. The metric is formally defined as:

$$\text{MRLR} = \frac{1}{U} \sum_{u=1}^{U} \frac{1}{\text{valid_rank}_u}$$

Together, *MRR* and *MRLR* provide complementary insights. *MRR* quantifies how accurately a model identifies the correct entity, while *MRLR* measures how often a model's top-ranked reasoning path overlaps a predefined *ROI*. We emphasise that *MRLR* quantifies agreement with the *ROI* templates rather than independently judged validity, and that it carries a different meaning across methods: for the non-interactive baselines, which never access the *ROI*s, *MRLR* is an independent check of whether their learned reasoning coincides with the *ROI* patterns. For *LiEr*, whose training preferences are *ROI*-derived, a high *MRLR*

[4]MINERVA: `https://github.com/shehzaadzd/MINERVA`, NTP/NTP-λ: `https://github.com/uclnlp/ntp`, NeuralLP: `https://github.com/fanyangxyz/Neural-LP`

[5]PyKEEN: `https://github.com/pykeen/pykeen`

(*S1*)	`locatedIn(Country, Continent)` ← `locatedIn(Country, Region)` ∧ `locatedIn(Region, Continent)`
(*S2*)	`locatedIn(Country X, Continent)` ← `neighborOf(Country X, Country Y)` ∧ `locatedIn(Country Y, Continent)`
(*S3*)	`locatedIn(Country X, Continent)` ← `neighborOf(Country X, Country Y)` ∧ `neighborOf(Country Y, Country Z)` ∧ `locatedIn(Country Z, Continent)` `locatedIn(Country X, Continent)` ← `neighborOf(Country X, Country Y)` ∧ `locatedIn(Country Y, Region)` ∧ `locatedIn(Region, Continent)`

Table 2: The reasoning patterns and *ROI*s a link prediction method has to learn to solve the *Countries S1*, *S2*, and *S3* tasks.

additionally reflects that the model recovered the templates that shaped its preferences. We retain *MRR* alongside *MRLR* precisely because *MRR* is not optimised through the *ROI*s and therefore provides an *ROI*-independent measure of predictive performance.

4.3 Benchmark Comparison

First, we aim to demonstrate that *LiEr* achieves reasoning capabilities on par with or better than existing state-of-the-art link prediction methods, despite relying solely on human preference feedback during training. To this end, *LiEr* is evaluated on the *Countries'* benchmark knowledge graphs (Bouchard et al., 2015; Rocktäschel & Riedel, 2017).

These benchmarks are specifically designed to assess multi-hop relational reasoning and are used in the literature for evaluating symbolic and neuro-symbolic link prediction methods (Bouchard et al., 2015; Rocktäschel & Riedel, 2017). Their structure and controlled complexity make them particularly suitable for isolating and analyzing reasoning behavior, independent of noise or ambiguity found in more realistic datasets.

The first evaluation focus on three reasoning tasks ($S1, S2, S3$), each requiring progressively more complex reasoning patterns to predict a country's continent accurately. These tasks serve as a controlled setting to assess whether *LiEr*, trained via human-aligned feedback, can match the performance of non-interactive models trained on direct supervision.

The *Countries* knowledge graph (Bouchard et al., 2015; Rocktäschel & Riedel, 2017) models 244 *Countries*, 23 `Regions`, and five `Continents`. The vertices are connected via edges with the symbols `locatedIn` and `neighborOf`. The knowledge graph provides three tasks with increasing complexity to assess the reasoning capabilities of link prediction methods. The goal of every task is to learn a reasoning pattern that predicts the `Continent` of a `Country`.

S1. In the first reasoning task (*S1*), all `locatedIn` edges directly connecting countries to continents are removed for the test set. To correctly answer queries in this setting, a model must infer continent membership through a transitive reasoning chain via intermediate regions (cf. Table 2). This setup isolates the model's ability to perform multi-hop inference without relying on direct links.

As shown in Table 3, *LiEr* achieves perfect *MRR* and *MRLR* scores, matching the performance of all neuro-symbolic baselines except for *NTP*, which marginally underperforms in this task. This result shows that *LiEr*, trained solely from human preferences, is capable of learning the required transitive reasoning pattern.

In contrast, all embedding-based methods fail on this task, exhibiting very low *MRR* values. This can be attributed to the semantic overload of the *locatedIn* relation, which connects both countries to regions and regions to continents. Embedding models like *TransE*, *DistMult*, and *ConvE* lack the structural expressiveness to differentiate such roles in context. Their learned representations collapse under this ambiguity, leading to mixed learning signals and severely degraded predictive performance. This pattern of failure among embedding-based models reappears in other reasoning-heavy benchmarks throughout the evaluation.

S2. In the second task (*S2*), direct `locatedIn` edges from countries to continents are again removed for the test set (cf. Table 2). However, in contrast to *S1*, the model must now infer continent membership by first identifying a neighboring country and then leveraging that country's `locatedIn` relationship to a continent.

This introduces additional complexity, as the model must navigate through a valid but less direct reasoning path that combines `neighborOf` and `locatedIn` relations.

As reported in Table 3 and Table 4, symbolic models such as *NeuralLP*, *NTP*, and *NTP-λ* perform best on this task, achieving near-perfect scores in both *MRR* and *MRLR*. These models reliably identify the valid reasoning paths that combine neighbor relations with region or continent information.

Once again, embedding-based models fail to learn the correct reasoning strategy, resulting in poor performance. The reasons mirror those observed in *S1*: due to the overloaded semantics of the `locatedIn` relation, these models are unable to distinguish between the different roles of entities.

LiEr and *MINERVA*, both path-based models, perform slightly worse than the neuro-symbolic baselines but still achieve high scores on both *MRR* and *MRLR*. Their relative underperformance can be explained by occasional failures to select the correct neighbor entity during the first hop. Nonetheless, their scores indicate that the intended reasoning pattern is learned in most cases and that both models can generate valid paths.

S3. The third task (*S3*) introduces the highest level of reasoning complexity (cf. Table 2). Here, models must infer the continent of a country through a three-hop reasoning chain, for example, first via a `neighborOf` relation to a neighboring country, then via a `locatedIn` relation to an intermediate region, and finally another `locatedIn` relation from the region to the correct continent. This setup is deliberately constructed such that reasoning patterns learned in *S1* and *S2*, which only require two hops, do not suffice to answer test queries.

As shown in Table 3 and Table 4, all embedding-based models again fail to produce meaningful predictions. This is consistent with earlier observations and further confirms their limitations in handling multi-hop relational semantics, especially under overloaded relations like `locatedIn`.

Interestingly, neuro-symbolic models such as *NTP*, *NTP-λ*, and *NeuralLP* exhibit a divergence between the two metrics. While their *MRR* is low, their *MRLR* remains high. This behavior is due to these models tendency to learn valid two-hop reasoning patterns during training, similar to those required for solving *S1* and *S2*. Although the reasoning patterns themselves are valid, and therefore rewarded under *MRLR*, they do not lead to correct predictions on the test set, explaining the performance drop in *MRR*.

MINERVA and *LiEr*, both path-based models, are the methods that are most successful in discovering the required three-hop reasoning pattern. *MINERVA* shows moderate performance, but with higher variance across runs. This is due to its reinforcement learning strategy being influenced by shorter two-hop paths that are rewarded during training, but which fail to generalize to the longer reasoning required in the test set. These misaligned paths often involve walking via `locatedIn` to a region, getting stuck without reaching a continent, and subsequently retracing steps, resulting in invalid trajectories and reduced *MRLR*.

By contrast, *LiEr* exhibits the highest performance on this task in both *MRR* and *MRLR*. The preference-based feedback mechanism enables *LiEr* to focus on valid three-hop paths explicitly aligned with the defined *ROI*s, avoiding the misleading reward signals that impact *MINERVA*. This result shows the advantage of interactive learning with structured feedback in guiding the model toward valid and generalizable reasoning patterns.

We note that the near-saturated *S1* and *S2* results should be read as sanity checks rather than as discriminating evidence between methods. Consistent with the role of *Countries* as the *MNIST* of our evaluation (Section 4), *S1* and *S2* verify only that a method captures the basic multi-hop reasoning mechanism. The results that actually separate the methods are *Countries S3*, where two-hop shortcuts no longer suffice and the more realistic *family*, *locations*, and *sports* benchmarks, where the methods diverge in reasoning validity; and the *Clever Hans Countries* benchmark, which is the central test of robustness to spurious correlations.

Across the three reasoning tasks in the *Countries* benchmark, *LiEr* consistently performs comparable with state-of-the-art models in terms of both predictive accuracy and reasoning validity. While relying entirely on preference-based feedback, *LiEr* identifies valid reasoning paths. These results validate that preference-driven learning can match the reasoning capabilities of supervised models in controlled settings. We now

turn to the less structured benchmarks (*family*, *locations*, and *sports*) to evaluate whether *LiEr*'s reasoning ability extends to more realistic knowledge graphs.

Model	Countries S1	Countries S2	Countries S3
ConvE	0.05 ± 0.00	0.05 ± 0.00	0.05 ± 0.01
TransE	0.03 ± 0.00	0.06 ± 0.00	0.03 ± 0.00
DistMult	0.04 ± 0.00	0.03 ± 0.00	0.04 ± 0.00
NTP	0.80 ± 0.10	0.95 ± 0.01	0.28 ± 0.00
NTP-λ	0.91 ± 0.07	0.98 ± 0.00	0.10 ± 0.01
NeuralLP	**1.00** ± 0.00	**1.00** ± 0.00	0.05 ± 0.00
Minerva	**1.00** ± 0.00	0.95 ± 0.00	0.49 ± 0.11
LiEr	**1.00** ± 0.00	0.90 ± 0.02	**0.85** ± 0.01

Table 3: *MRR* (↑) metric and variance for the three tasks of the *Countries* knowledge graph. Mean and variance from 10 training runs.

Model	Countries S1	Countries S2	Countries S3
NTP	0.77 ± 0.15	**1.00** ± 0.00	**1.00** ± 0.00
NTP-λ	**1.00** ± 0.00	**1.00** ± 0.00	**1.00** ± 0.00
NeuralLP	**1.00** ± 0.00	**1.00** ± 0.00	**1.00** ± 0.00
Minerva	**1.00** ± 0.00	0.95 ± 0.00	0.33 ± 0.01
LiEr	**1.00** ± 0.00	0.93 ± 0.01	0.88 ± 0.01

Table 4: *MRLR* (↑) metric and variance for the three tasks of the *Countries* knowledge graph. Mean and variance from 10 training runs.

Family. The *family* knowledge graph (Yang et al., 2017)[6] is designed to test relational reasoning under a fixed set of logically interdependent relation types. It contains 12 relationship types, such as `aunt`, `brother`, `daughter`, `father`, `niece`, and `uncle`, distributed across 3007 entities. Each relation can be reconstructed using valid multi-hop compositions of other relations. For example:

$$\texttt{daughter(X,Y)} \leftarrow \texttt{wife(X,Z)} \wedge \texttt{daughter(Z,Y)}$$
$$\texttt{uncle(X,Y)} \leftarrow \texttt{brother(X,Z)} \wedge \texttt{father(Z,Y)}$$
$$\texttt{aunt(X,Y)} \leftarrow \texttt{niece(Y,X)}$$

These strict symbolic dependencies make *family* an ideal benchmark for defining a complete set of *ROI*s and evaluating models on their capacity for valid logical reasoning.

Embedding-based models such as *TransE*, *ConvE*, and *DistMult* fail to predict missing links in this knowledge graph, with *MRR* scores close to zero (Table 5). This failure is due to structural redundancy in the graph. Most nodes have similarly shaped neighborhoods (parents, siblings, extended family), which leads to high embedding similarity across distinct entities. The result is oversmoothing (Hoseinnia et al., 2025), making it difficult for these models to differentiate entities based on vector representations alone.

Surprisingly, neuro-symbolic models such as *NTP* and *NTP-λ* also perform poorly. An inspection of the rules they learn reveals that some valid rules are extracted, for example, `aunt(X,Y)` ← `niece(Y,X)`, and `daughter(X,Y)` ← `niece(X,Z)` ∧ `brother(Z,Y)`. However, *NTP* struggles with inconsistent directionality of certain relations in the data. For example, the `father` relation may point from parent to child in one instance, and from child to parent in another. Such bidirectional semantics complicate rule generalization for *NTP*. *NeuralLP* achieves high performance on both *MRR* and *MRLR*. It learns robustly valid reasoning patterns under directionality ambiguities. Among the path-based models, both *MINERVA* and *LiEr* show moderate performance, with *LiEr* slightly outperforming *MINERVA* on both metrics. Both models are able to discover valid paths, but struggle with the same directionality inconsistencies.

Overall, the *Family* dataset has two core challenges: oversmoothing in embedding-based methods and directional ambiguity in symbolic reasoning, making it a complex graph for link prediction. *LiEr* struggles with those challenges. However, it shows especially in terms of *MRLR* improvements to methods like *NTP* and *Minerva*.

Locations. The *locations* benchmark is a subset of relations from the *Never-Ending Language Learning (NELL)* knowledge graph (Mitchell et al., 2018). It contains 445 entities spanning the classes `state`, `city`, `country`, and `capital`, and includes five relation types: `concept_citycapitalofcountry`, `concept_citylocatedincountry`, `concept_citylocatedinstate`, `concept_statehascapital`, and `concept_statelocatedincountry`. Each relation can be reconstructed through valid multi-hop reasoning chains. For example:

[6]Dataset available at: `https://github.com/fanyangxyz/Neural-LP/tree/master/datasets/family`

$$\texttt{concept_citylocatedincountry(X,Y)} \leftarrow \texttt{concept_citylocatedinstate(X,Z)} \wedge \texttt{concept_statelocatedincountry(Z,Y)}$$
$$\texttt{concept_statehascapital(X,Y)} \leftarrow \texttt{concept_statelocatedincountry(X,Z)} \wedge \texttt{concept_citycapitalofcountry(Y,Z)}$$

These valid relational dependencies provide *ROI*s against which reasoning can be assessed.

As shown in Table 5, embedding-based models such as *TransE*, *ConvE*, and *DistMult* again fail to achieve meaningful performance. In this case, the issue is not oversmoothing, as in the *family* dataset, but the limitation of low-dimensional embeddings to capture the structured, multi-hop reasoning patterns required for correct predictions.

Neuro-symbolic models perform substantially better. *NTP*, *NTP-λ*, *MINERVA*, and *LiEr* achieve comparable *MRR* scores, with *LiEr* achieving the highest *MRLR*. This indicates that *LiEr* produces reasoning paths more consistently aligned with valid human-understandable chains. For instance, a frequently observed pattern learned by *LiEr* is:

$$\texttt{concept_citylocatedincountry(X,Y)} \leftarrow \texttt{concept_statehascapital(X,Z)} \wedge \texttt{concept_statelocatedincountry(Z,Y)}$$

This path reflects correct domain reasoning: a city may be identified as the capital of a state, and the state can then be linked to its country.

NeuralLP, however, achieves nearly perfect performance on *MRR*, while its *MRLR* collapses to zero (cf. Table 6). Inspection of its learned rules reveals the cause. For example, NeuralLP frequently induces homogeneous transitive chains such as:

$$\texttt{concept_citylocatedincountry(C,A)} \leftarrow \texttt{concept_citylocatedincountry(B,A)} \wedge \texttt{concept_citylocatedincountry(C,B)}$$

Although these chains provide high predictive accuracy on the dataset, they are invalid: a city cannot be located in another city, nor does such chaining reflect meaningful geographic relations. Thus, *NeuralLP* achieves high predictive power by exploiting structural regularities, but fails to produce reasoning paths aligned with human intuition.

Overall, while *LiEr* does not outperform all models in terms of predictive accuracy, it has the highest reasoning validity among the methods. This shows *LiEr*'s strength as an interactive, preference-guided approach that balances accuracy with alignment to human-understandable reasoning.

Sports. The *sports* knowledge graph is another subset of the *NELL* knowledge graph (Mitchell et al., 2018). It comprises 1039 entities from categories such as `athletes`, `sports teams`, `universities`, `sports leagues`, and `organizations`, and includes four relation types: `concept_athleteledsportsteam`, `concept_athleteplaysforteam`, `concept_coachesteam`, and `concept_personbelongstoorganization`. Each of these relation types can be reconstructed through reasoning patterns. For example:

$$\texttt{concept_personbelongstoorganization(X,Y)} \leftarrow \texttt{concept_coachesteam(X,Y)}$$

Such patterns can be used to define *ROI*s for preference-aligned training and evaluation.

As shown in Table 5, embedding-based methods such as *TransE*, *ConvE*, and *DistMult* once again fail to capture meaningful structure in the data. The failure is consistent with their performance on the *locations* knowledge graph. Unlike noisy, large-scale graphs where embeddings can pick up on distributional cues, the *sports* knowledge graph requires explicit, structured reasoning to identify correct links. This reveals a core weakness of embedding-based models in reasoning-heavy scenarios.

The neuro-symbolic models perform substantially better. *NTP*, *NTP-λ*, *MINERVA*, and *LiEr* all achieve strong *MRR* and *MRLR* scores, with *NTP-λ* again showing an advantage in reasoning validity.

However, *NeuralLP*, despite its strong *MRR*, once again learns reasoning patterns that are predictive but invalid. For example, it frequently learns transitive chains within a single relation type:

$$\texttt{concept_personbelongstoorganization(X,Y)} \leftarrow \texttt{concept_personbelongstoorganization(X,Z)} \wedge \texttt{concept_personbelongstoorganization(Z,Y)}$$

While such rules may help capture statistical regularities in the dataset, they do not reflect valid reasoning paths and do not contribute to human-aligned interpretability and trust in the prediction, as reflected in the low *MRLR* score for *NeuralLP* on this benchmark.

In summary, *LiEr* demonstrates reasoning capabilities comparable to the best-performing neuro-symbolic methods on more realistic, reasoning-centric knowledge graphs such as *family*, *locations*, and *sports*. However, the strength of *LiEr* lies in its robustness to spurious correlations and its ability to align with human understanding of valid reasoning. To evaluate this property, the final benchmark considers the *Clever Hans Countries* knowledge graph.

Model	family	locations	sports
ConvE	0.04 ± 0.00	0.02 ± 0.00	0.01 ± 0.00
TransE	0.03 ± 0.00	0.02 ± 0.00	0.01 ± 0.00
DistMult	0.03 ± 0.00	0.01 ± 0.00	0.01 ± 0.00
NTP	0.03 ± 0.00	0.58 ± 0.00	0.72 ± 0.00
NTP-λ	0.03 ± 0.00	0.63 ± 0.00	0.75 ± 0.00
NeuralLP	**0.70** ± 0.00	**0.99** ± 0.00	**0.84** ± 0.00
Minerva	0.24 ± 0.00	0.49 ± 0.00	0.79 ± 0.00
LiEr	0.28 ± 0.00	0.52 ± 0.00	0.79 ± 0.00

Table 5: *MRR* (↑) metric and variance for the *family*, *sports* and *locations* knowledge graphs. Mean and variance from 10 training runs.

Model	family	locations	sports
NTP	0.06 ± 0.00	0.34 ± 0.02	0.75 ± 0.12
NTP-λ	0.07 ± 0.00	0.29 ± 0.01	**0.88** ± 0.05
NeuralLP	**0.95** ± 0.00	0.00 ± 0.00	0.01 ± 0.00
Minerva	0.68 ± 0.00	0.34 ± 0.00	0.76 ± 0.00
LiEr	0.77 ± 0.00	**0.37** ± 0.00	0.79 ± 0.00

Table 6: *MRLR* (↑) metric and variance for the *family*, *sports* and *locations* knowledge graphs. Mean and variance from 10 training runs.

4.4 Evaluating LiEr's Alignment with Valid Reasoning

This section evaluates whether *LiEr* is capable of aligning its reasoning with human expectations, even when spurious but predictive patterns dominate the training data. To that end, we introduce the *Clever Hans Countries* knowledge graph, a modification of the *Countries S3* knowledge graph designed to simulate a scenario where spurious correlations exist during training but do not hold at test time.

Model	Clever Hans Countries
ConvE	0.39 ± 0.02
TransE	0.35 ± 0.01
DistMult	0.29 ± 0.03
NTP	0.08 ± 0.00
NTP-λ	0.08 ± 0.03
NeuralLP	0.08 ± 0.00
Minerva	0.23 ± 0.07
LiEr	**0.86** ± 0.08

Table 7: *MRR* (↑) metric and variance for the *Clever Hans Countries* knowledge graph. Mean and variance from 10 training runs.

Model	Clever Hans Countries
NTP	0.14 ± 0.13
NTP-λ	0.05 ± 0.05
Neural LP	0.00 ± 0.00
Minerva	0.22 ± 0.07
LiEr	**0.86** ± 0.08

Table 8: *MRLR* (↑) metric and variance for the *Clever Hans Countries* knowledge graph. Mean and variance from 10 training runs.

First, to increase structural clarity, all `locatedIn` relations in *Countries S3* are refactored into disjoint relation types: `inRegion`, `regionInContinent`, and `countryInContinent`. In addition, a spurious relation,

consumesPepperFrom, is introduced, linking each country in the training and validation sets directly to its correct continent. However, in the test set, this relation systematically points to incorrect continents. As a result, link prediction models that rely on the consumesPepperFrom relation will appear to perform well on the validation set, but generalize poorly to the test set, mirroring the classic Clever Hans effect (Lapuschkin et al., 2019).

Table 7 shows that *LiEr* achieves a significantly higher *MRR* score than all other models, with low variance across runs. Its reasoning is also valid, as reflected in the similarly high *MRLR* score (cf. Table 8). This is due to *LiEr*'s preference-based feedback never rewarding reasoning paths that involve the consumesPepperFrom relation, preventing the model from learning the spurious pattern in the first place.

In contrast, symbolic models such as *NTP*, *NTP-*λ, and *NeuralLP* perform poorly. Manual inspection reveals that these models learn reasoning patterns involving the spurious consumesPepperFrom relation. While these patterns are structurally valid under the training distribution, they fail catastrophically at test time, causing the observed drop in *MRR* and *MRLR*. This illustrates the core motivation of *LiEr*: rule-based systems are susceptible to misleading supervision signals (Marconato et al., 2023) and can fail to distinguish between valid and invalid reasoning if no alignment with human expectations is enforced.

MINERVA partially recovers from the spurious signal, achieving moderate performance, though both its *MRR* and *MRLR* remain substantially lower than *LiEr*'s. Its path-based strategy tends to prioritize paths involving the misleading consumesPepperFrom relation. This results in paths that end at incorrect nodes.

Table 9 provides a qualitative comparison of reasoning patterns learned by different methods. *MINERVA*, which rewards purely based on output correctness, assigns high probability to short reasoning paths involving the spurious consumesPepperFrom relation. While these paths appear effective during training and validation, they fail to generalise to the test set, as the relation does not hold under the true distribution. A similar failure mode is observed in neuro-symbolic models such as *NTP* and *NeuralLP*, which also focus on learning rules that correlate strongly with the training data. These models also tend to generate misleading rules that rely on the spurious relation.

In contrast, *LiEr*, trained with preference-based feedback, consistently favours reasoning paths that align with human understanding of valid reasoning, such as traversing through a neighbor, identifying the correct region, and inferring the continent via a three-hop relational chain. These patterns are more complex and less frequent in the training set but are aligned with the intended semantics of the countryInContinent relation. Because the oracle's preferences actively reinforce such paths, *LiEr* avoids the spurious shortcut and maintains high accuracy at test time.

These results confirm that *LiEr* is able, via preference-based feedback, to align its reasoning with human understanding of valid reasoning, and this results in superior predictive performance when spurious correlations are present in the data. Thus, *LiEr*'s value lies in generalizing robustly in deceptive environments. We further note that *LiEr*'s advantage is not an artefact of asymmetric inputs. On the standard benchmarks *LiEr* is on par with the baselines on predictive accuracy (Tables 3 and 5). It outperforms the other methods on *Clever Hans Countries*, where the preference signal is what suppresses the spurious shortcut. This is the intended effect of *LiEr*.

4.5 Remarks on the Collection of Preference-based Feedback

This section provides remarks on the preference collection and *LiEr*'s robustness against erroneous preferences. The preference-based feedback from a human oracle is collected via the command line (cf. Figure 4).

The command-line interface (Figure 4) was used to collect real human preferences on the *Countries S1* task, to verify that the preference-collection procedure operates with a human annotator and not only with the simulated oracle. In that setting, *LiEr* learned the required transitive reasoning pattern ($MRR = 1$) with 12 pairwise preferences collected over five training epochs. Collecting preferences via pairwise comparison proved readily accessible; in the general-knowledge *Countries* domain, a single annotator can express on the order of 70 preferences in ten minutes, though time-to-preference varies with reasoning-path length and the annotator's domain expertise.

MINERVA	0.91 countryInContinent(Israel, Africa) ← consumesPepperFrom(Israel, Africa) 0.07 countryInContinent(Israel, Africa) ← neighborOf(Israel, Egypt) ∧ consumesPepperFrom(Egypt, Africa) 0.02 countryInContinent(Israel, Asia) ← neighborOf(Israel, Lebanon) ∧ inRegion(Lebanon, West Asia) ∧ regionInContinent(West Asia, Asia)
NeuralLP	0.72 countryInContinent(Country, Continent) ← consumesPepperFrom(Country, Continent) 0.18 countryInContinent(Country, Continent) ← neighborOf(Country, Country) ∧ consumesPepperFrom(Country, Continent) 0.08 countryInContinent(Country, Continent) ← neighborOf(Country, Country) ∧ neighborOf(Country, Country) ∧ consumesPepperFrom(Country, Continent)
LiEr	0.78 countryInContinent(Israel, Asia) ← neighborOf(Israel, Lebanon) ∧ inRegion(Lebanon, West Asia) ∧ regionInContinent(West Asia, Asia) 0.17 countryInContinent(Israel, Asia) ← inRegion(Israel, West Asia) ∧ regionInContinent(West Asia, Asia) 0.04 countryInContinent(Israel, Asia) ← neighborOf(Israel, Jordan) ∧ inRegion(Jordan, West Asia) ∧ regionInContinent(West Asia, Asia)

Table 9: Exemplary top three reasoning paths and their predicted probabilities for the query `(Israel, countryInContinent, ?)`, comparing the behavior of *MINERVA* , *NeuralLP*, and *LiEr*. Reasonings are extracted from representative training runs.

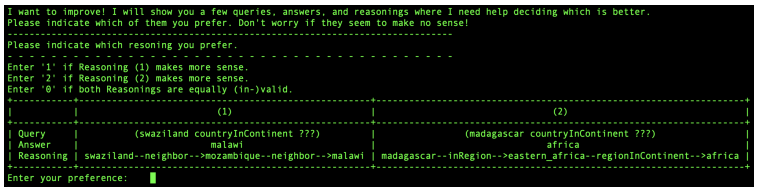

Figure 4: A command line interface for collecting preferences. *LiEr* prompts the human oracle to express preferences over reasoning pairs.

The pairwise comparison proved to be easily accessible for humans while testing *LiEr*. It enables humans to effortlessly communicate preferences to *LiEr*. However, the concrete labeling time varies with the length of the reasoning paths and the human expertise in the domain of interest.

It is particularly noteworthy that expressing preferences over incorrect pairs of reasoning paths helps *LiEr* to learn valid reasoning paths. Especially in the beginning of learning more complex tasks like *Countries S3* it proves beneficial to prefer the reasoning of two incorrect reasoning paths that includes the `locatedIn` relation. This leads *LiEr* to expand its reasoning paths towards `Regions`, increasing the likelihood of sampling paths that take the `locatedIn` relation from `Regions` to `Continents`.

Erroneous preferences. The results reported in this section constitute that *LiEr* learns valid reasoning from human feedback. However, even domain experts make errors while expressing preferences. They may prefer by accident or because of erroneous knowledge, invalid or incorrect reasoning paths over valid reasoning paths. Figure 5 shows how robust *LiEr* handles erroneous preferences.

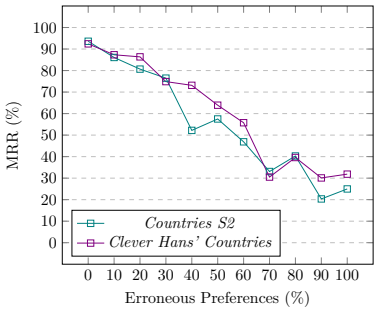

Figure 5: The *MRR* of *LiEr* on *Countries S2* and *Clever Hans Countries* with an increasing amount of erroneous preferences. The reported numbers are the mean of 20 training runs per erroneous preferences step. Reasoning pairs for erroneous preferences are selected uniformly at random.

The *Clever Hans Countries* and *Countries S2* tasks are selected for the test. They represent one challenging and one mediocre-challenging reasoning task. *LiEr* shows a minor drop in performance from 0% to 30% erroneous preferences. Meanwhile, a heavy increase in variance is observable, up to 19.42 points at 30% for *Clever Hans Countries* and 11.57 points at 40% for *Countries S2*. After that, the performance drops, and the variance decreases slowly. The results illustrate that *LiEr* is more dependent on correct feedback the more complex a reasoning task gets.

The test assumes that the expert is wrong at random. In reality, an expert is not wrong at random but biased in its error. This may result in *LiEr* being more or less robust against erroneous preferences, depending on the bias of the error.

This section showed *LiEr*'s on-par reasoning capabilities with state-of-the-art link prediction methods. It demonstrated that *LiEr* learns valid and correct reasoning patterns from preference-based human feedback. In addition, the section demonstrated that valid reasoning patterns lead to increased performance in knowledge graphs with high risks to induce the *Clever Hans* biases.

4.6 Discussion and Limitations

The results across all benchmarks demonstrate that *LiEr* provides a viable approach for incorporating preference-based supervision into link prediction tasks. *LiEr* proves especially useful in settings where the goal is to predict missing link and to guide the reasoning process, whether due to constraints on valid reasoning, domain-specific heuristics, or the presence of spurious signals. It enables learning from pairwise preferences over reasoning paths, allowing for interactive, human-in-the-loop training without requiring fully labeled data.

This makes *LiEr* a strong candidate for scenarios where knowledge graph completion needs to align with external objectives (e.g., legal reasoning, clinical inference, or pedagogy-driven tutoring systems) rather than simply fitting observed graph structure. In particular, *LiEr* offers an interface for aligning reasoning with user-defined or domain-specific knowledge about semantic concepts, through its preference-based reward mechanism.

However, there are practical challenges that come with relying on preference feedback instead of supervised triples. The most significant limitation lies in collecting sufficient high-quality feedback. In real-world use cases, this typically requires human input, which can be time-consuming and costly. A concern is how *LiEr* applies to open-domain knowledge graphs where valid reasoning patterns are incomplete, ambiguous, or cannot be written down as templates. We note that this is the setting *LiEr* is designed for. The method does not ask humans to define valid reasoning templates, but to express pairwise preferences over concrete reasoning paths. As argued in Section 3.2.3, domain experts frequently cannot verbalise valid reasoning rules, yet can readily judge whether a presented path is appropriate. The inability to articulate templates is therefore the motivation for the preference-based approach and not a barrier to it. In open-domain use one elicits pairwise human judgements directly, in place of authoring *ROI*s. We are nonetheless careful about the boundary of this claim. *LiEr* addresses settings where valid reasoning is hard to articulate but can still be recognised, not settings where validity is genuinely unknown or undefined, since reliable preferences are then unavailable. For reproducibility our evaluation uses knowledge graphs with well-defined semantics, where *ROI*s can simulate the oracle. This leaves open-domain feasibility, where preferences must be elicited from humans and validity may be partial or contested, as a question we do not resolve empirically here, besides Figure 5. Future work may reduce the human effort by automating preference generation with large language models.

Another important factor is the sensitivity of *LiEr*'s performance to the initial feedback pairs. If the first rounds of feedback include poorly chosen comparisons, the model may get pushed into invalid reasoning regions of the search space, making recovery difficult. In practice, we found that early-phase instability could be mitigated by terminating runs with uninformative comparisons during the warm-up phase. This reduced overall variance and improved robustness.

On real-world benchmarks, we also observed that performance improved when *LiEr* was first pretrained using standard correctness-based supervision before switching to preference-based training. This hybrid

approach allowed the model to develop an initial understanding of graph topology, which was then refined using human-aligned feedback.

In summary, *LiEr* shows strong reasoning capabilities and interpretable behavior under preference-guided learning, particularly in settings where valid reasoning matters more than purely predictive performance. The method is most suitable in contexts where structured feedback is available or can be efficiently elicited and where aligning predictions with human intuition or domain knowledge is critical.

5 Conclusion

This paper introduced *LiEr*, a human-in-the-loop link prediction method designed to learn valid reasoning from preference-based feedback. Unlike conventional link prediction models that rely solely on observed graph patterns, *LiEr* allows users to guide and align the model's reasoning behavior with human notions of validity. It is the first method to enable such interactive training in a multi-hop link prediction setting.

LiEr was evaluated across seven benchmark knowledge graphs, ranging from synthetic toy knowledge graphs (*Countries*) to more realistic, reasoning-heavy domains such as *family*, *sports*, and *locations*, as well as the deliberately biased *Clever Hans Countries* knowledge graph. The results demonstrate that *LiEr* performs on par with state-of-the-art neuro-symbolic link prediction models in terms of predictive accuracy, while consistently producing reasoning paths that better align with human-understandable patterns. In particular, *LiEr* remained robust on the *Clever Hans* dataset, where preference-based feedback effectively shielded it from spurious training correlations that misled other models.

LiEr is particularly suited for applications where model reasoning needs to be controllable, interpretable, or auditable. This includes domains where valid reasoning is critical, where the target relation does not exist in the training data, or where alignment with expert domain knowledge is required.

Future work should evaluate *LiEr* on large-scale, real-world reasoning tasks such as biomedical link prediction (Szklarczyk et al., 2020), scientific hypothesis generation (Besold et al., 2025), and safety-critical knowledge graphs in automotive and manufacturing (Wehner et al., 2022; Bahr et al., 2025a). In addition, future research may explore the use of *LiEr*'s preference-based reward to fine-tune existing path-based models such as *MINERVA* (Das et al., 2018) or *CURL* (Zhang et al., 2022), to align pre-trained models with domain-specific reasoning expectations.

LiEr empowers users to guide link prediction beyond pattern recognition. It enables alignment between machine-learned reasoning and human judgment, putting the human back in the loop.

References

Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '19, pp. 2623–2631, New York, NY, USA, 2019. Association for Computing Machinery. ISBN 9781450362016. doi: 10.1145/3292500.3330701. URL `https://doi.org/10.1145/3292500.3330701`.

Mehdi Ali, Max Berrendorf, Charles Tapley Hoyt, Laurent Vermue, Sahand Sharifzadeh, Volker Tresp, and Jens Lehmann. PyKEEN 1.0: A Python Library for Training and Evaluating Knowledge Graph Embeddings. *Journal of Machine Learning Research*, 22(82):1–6, 2021. URL `http://jmlr.org/papers/v22/20-825.html`.

Ibrahim Assem, Andrzej Skowronski, and Daniel Simson. *Elements of the Representation Theory of Associative Algebras: Techniques of Representation Theory*, volume 1 of *London Mathematical Society Student Texts*. Cambridge University Press, 2006. doi: 10.1017/CBO9780511614309.

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. Layer normalization, 2016.

Lukas Bahr, Lucas Poßner, Konstantin Weise, Sophie Gröger, and Rüdiger Daub. Sensitivity analysis of image classification models using generalized polynomial chaos, 2025a. URL https://arxiv.org/abs/2506.18751.

Lukas Bahr, Christoph Wehner, Judith Wewerka, José Bittencourt, Ute Schmid, and Rüdiger Daub. Knowledge graph enhanced retrieval-augmented generation for failure mode and effects analysis. *Journal of Industrial Information Integration*, 45:100807, 2025b. ISSN 2452-414X. doi: https://doi.org/10.1016/j.jii.2025.100807. URL https://www.sciencedirect.com/science/article/pii/S2452414X25000317.

Richard Bellman. A markovian decision process. *Indiana Univ. Math. J.*, 6:679–684, 1957. ISSN 0022-2518.

James Bergstra, Rémi Bardenet, Yoshua Bengio, and Balázs Kégl. Algorithms for hyper-parameter optimization. In *Proceedings of the 25th International Conference on Neural Information Processing Systems*, NIPS'11, pp. 2546–2554, Red Hook, NY, USA, 2011. Curran Associates Inc. ISBN 9781618395993.

Tarek R. Besold, Uchenna Akujuobi, Samy Badreddine, Jihun Choi, Hatem Elshazly, Frederick Gifford, Chrysa Iliopoulou, Kana Maruyama, Kae Nagano, Pablo Sanchez Martin, Thiviyan Thanapalasingam, Alessandra Toniato, and Christoph Wehner. Literature-based hypothesis generation: Predicting the evolution of scientific literature to support scientists. In *AI4X 2025 International Conference*, 2025. URL https://openreview.net/forum?id=e2FXu9lsYF.

Rajarshi Bhowmik and Gerard de Melo. Explainable link prediction for emerging entities in knowledge graphs. In Jeff Z. Pan, Valentina Tamma, Claudia d'Amato, Krzysztof Janowicz, Bo Fu, Axel Polleres, Oshani Seneviratne, and Lalana Kagal (eds.), *The Semantic Web – ISWC 2020*, pp. 39–55, Cham, 2020. Springer International Publishing. ISBN 978-3-030-62419-4.

Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. Translating embeddings for modeling multi-relational data. In C.J. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K.Q. Weinberger (eds.), *Advances in Neural Information Processing Systems*, volume 26, pp. 1–9. Curran Associates, Inc., 2013. URL https://proceedings.neurips.cc/paper_files/paper/2013/file/1cecc7a77928ca8133fa24680a88d2f9-Paper.pdf.

Guillaume Bouchard, Sameer Singh, and Théo Trouillon. On approximate reasoning capabilities of low-rank vector spaces. In *Knowledge Representation and Reasoning: Integrating Symbolic and Neural Approaches, AAAI Spring Symposium Series*, pp. 1–4, 2015.

Ralph Allan Bradley and Milton E. Terry. Rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika*, 39(3/4):324–345, 1952. ISSN 00063444. URL http://www.jstor.org/stable/2334029.

Ron Brinkmann. *The Art and Science of Digital Compositing, Second Edition: Techniques for Visual Effects, Animation and Motion Graphics (The Morgan Kaufmann Series in ... Morgan Kaufmann Series in Computer Graphics)*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2 edition, 2008. ISBN 0123706386.

Yang Chen, Daisy Zhe Wang, and Sean Goldberg. ScaLeKB: Scalable learning and inference over large knowledge bases. *The VLDB Journal*, 25:893–918, 12 2016. ISSN 1066-8888. doi: 10.1007/s00778-016-0444-3. URL https://doi.org/10.1007/s00778-016-0444-3.

Kewei Cheng, Nesreen Ahmed, and Yizhou Sun. Neural compositional rule learning for knowledge graph reasoning. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=F8VKQyDgRVj.

Brian Christian. *The Alignment Problem: Machine Learning and Human Values*. WW Norton, 2020. ISBN 9780393635829. URL https://books.google.de/books?id=VmJIzQEACAAJ.

Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 30,

pp. 1–9. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper/2017/file/d5e2c0adad503c91f91df240d0cd4e49-Paper.pdf.

Irving M. Copi. Introduction to logic. *Revue de Métaphysique et de Morale*, 59(3):344–345, 1954.

Rajarshi Das, Shehzaad Dhuliawala, Manzil Zaheer, Luke Vilnis, Ishan Durugkar, Akshay Krishnamurthy, Alex Smola, and Andrew McCallum. Go for a walk and arrive at the answer: Reasoning over paths in knowledge bases using reinforcement learning. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*, pp. 1–18. OpenReview.net, 2018. URL https://openreview.net/forum?id=Syg-YfWCW.

Li Deng. The mnist database of handwritten digit images for machine learning research. *IEEE Signal Processing Magazine*, 29(6):141–142, 2012.

Tim Dettmers, Pasquale Minervini, Pontus Stenetorp, and Sebastian Riedel. Convolutional 2d knowledge graph embeddings. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence and Thirtieth Innovative Applications of Artificial Intelligence Conference and Eighth AAAI Symposium on Educational Advances in Artificial Intelligence*, AAAI'18/IAAI'18/EAAI'18. AAAI Press, 2018. ISBN 978-1-57735-800-8.

Thomas G Dietterich et al. Ensemble learning. *The handbook of brain theory and neural networks*, 2(1): 110–125, 2002.

M.J. Evans, T. Swartz, and A.P.D.M.S.T. Swartz. *Approximating Integrals Via Monte Carlo and Deterministic Methods*. Oxford statistical science series. Oxford University Press, 2000. ISBN 9780198502784. URL https://books.google.de/books?id=SwbomAEACAAJ.

Norbert Fuhr. Some common mistakes in ir evaluation, and how they can be avoided. *SIGIR Forum*, 51 (3):32–41, feb 2018. ISSN 0163-5840. doi: 10.1145/3190580.3190586. URL https://doi.org/10.1145/3190580.3190586.

Iason Gabriel. Artificial intelligence, values, and alignment. *Minds and Machines*, 30(3):411–437, Sep 2020. ISSN 1572-8641. doi: 10.1007/s11023-020-09539-2. URL https://doi.org/10.1007/s11023-020-09539-2.

Joseph Gallian. A dynamic survey of graph labeling. *Electron. J. Combin., Dynamic Surveys*, 19, 11 2000.

Luis Galárraga, Christina Teflioudi, Katja Hose, and Fabian M Suchanek. Fast rule mining in ontological knowledge bases with amie+. *The VLDB Journal*, 24:707–730, 2015. ISSN 0949-877X. doi: 10.1007/s00778-015-0394-1. URL https://doi.org/10.1007/s00778-015-0394-1.

Ian J. Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, Cambridge, MA, USA, 2016. http://www.deeplearningbook.org.

L. Guan, Mudit Verma, Sihang Guo, Ruohan Zhang, and Subbarao Kambhampati. Widening the pipeline in human-guided reinforcement learning with explanation and context-aware data augmentation. In *Neural Information Processing Systems*, 2020.

J. Hammersley. *Monte Carlo Methods*. Monographs on Statistics and Applied Probability. Springer Netherlands, 2013. ISBN 9789400958197. URL https://books.google.de/books?id=3rDvCAAAQBAJ.

Frank. Harary, Robert Z (Robert Zane) Norman, and Dorwin Cartwright. *Structural models: an introduction to the theory of directed graphs*. Wiley, 1965.

Aidan Hogan, Eva Blomqvist, Michael Cochez, Claudia D'amato, Gerard De Melo, Claudio Gutierrez, Sabrina Kirrane, José Emilio Labra Gayo, Roberto Navigli, Sebastian Neumaier, Axel-Cyrille Ngonga Ngomo, Axel Polleres, Sabbir M. Rashid, Anisa Rula, Lukas Schmelzeisen, Juan Sequeda, Steffen Staab, and Antoine Zimmermann. Knowledge graphs. *ACM Computing Surveys*, 54(4):1–37, may 2022. doi: 10.1145/3447772. URL https://doi.org/10.1145%2F3447772.

Fateme Hoseinnia, Mehdi Ghatee, and Mostafa Haghir Chehreghani. Mitigating over-smoothing in graph neural networks for node classification through adaptive early embedding and biased dropedge procedures. *Knowledge-Based Systems*, 320:113615, 2025. ISSN 0950-7051. doi: https://doi.org/10.1016/j.knosys.2025.113615. URL https://www.sciencedirect.com/science/article/pii/S0950705125006616.

Filip Ilievski, Barbara Hammer, Frank van Harmelen, Benjamin Paassen, Sascha Saralajew, Ute Schmid, Michael Biehl, Marianna Bolognesi, Xin Luna Dong, Kiril Gashteovski, Pascal Hitzler, Giuseppe Marra, Pasquale Minervini, Martin Mundt, Axel-Cyrille Ngonga Ngomo, Alessandro Oltramari, Gabriella Pasi, Zeynep G. Saribatur, Luciano Serafini, John Shawe-Taylor, Vered Shwartz, Gabriella Skitalinskaya, Clemens Stachl, Gido M. van de Ven, and Thomas Villmann. Aligning generalization between humans and machines. *Nature Machine Intelligence*, 7(9):1378–1389, September 2025. ISSN 2522-5839. doi: 10.1038/s42256-025-01109-4. URL https://www.nature.com/articles/s42256-025-01109-4. Publisher: Nature Publishing Group.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. URL http://arxiv.org/abs/1412.6980.

W.B̃radley Knox and Peter Stone. Augmenting reinforcement learning with human feedback. In *ICML 2011 Workshop on New Developments in Imitation Learning*, July 2011.

Sebastian Lapuschkin, Stephan Wäldchen, Alexander Binder, Grégoire Montavon, Wojciech Samek, and Klaus-Robert Müller. Unmasking clever hans predictors and assessing what machines really learn. *Nature Communications*, 10(1):1096, Mar 2019. ISSN 2041-1723. doi: 10.1038/s41467-019-08987-4. URL https://doi.org/10.1038/s41467-019-08987-4.

Kimin Lee, Laura M Smith, and Pieter Abbeel. Pebble: Feedback-efficient interactive reinforcement learning via relabeling experience and unsupervised pre-training. In Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 6152–6163. PMLR, 18–24 Jul 2021. URL https://proceedings.mlr.press/v139/lee21i.html.

Xi Victoria Lin, Richard Socher, and Caiming Xiong. Multi-hop knowledge graph reasoning with reward shaping. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 3243–3253, Brussels, Belgium, October-November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1362. URL https://aclanthology.org/D18-1362.

Shiteng Lu, Xinqiang Li, Wenqi Zhang, and Huanling Tang. RoEMF: rotational embedding multimodal fusion for link prediction. *Knowledge and Information Systems*, March 2025. ISSN 0219-3116. doi: 10.1007/s10115-025-02386-6. URL https://doi.org/10.1007/s10115-025-02386-6.

Saaduddin Mahmud, Mason Nakamura, and Shlomo Zilberstein. Maple: A framework for active preference learning guided by large language models, 2024. URL https://arxiv.org/abs/2412.07207.

Emanuele Marconato, Stefano Teso, Antonio Vergari, and Andrea Passerini. Not all neuro-symbolic concepts are created equal: analysis and mitigation of reasoning shortcuts. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, Red Hook, NY, USA, 2023. Curran Associates Inc.

Christian Meilicke, Melisachew Wudage Chekol, Patrick Betz, Manuel Fink, and Heiner Stuckeschmidt. Anytime bottom-up rule learning for large-scale knowledge graph completion. *The VLDB Journal*, 33(1): 131–161, January 2024. ISSN 0949-877X. doi: 10.1007/s00778-023-00800-5. URL https://doi.org/10.1007/s00778-023-00800-5.

T. Mitchell, W. Cohen, E. Hruschka, P. Talukdar, B. Yang, J. Betteridge, A. Carlson, B. Dalvi, M. Gardner, B. Kisiel, J. Krishnamurthy, N. Lao, K. Mazaitis, T. Mohamed, N. Nakashole, E. Platanios, A. Ritter, M. Samadi, B. Settles, R. Wang, D. Wijaya, A. Gupta, X. Chen, A. Saparov, M. Greaves, and J. Welling. Never-ending learning. *Commun. ACM*, 61(5):103–115, April 2018. ISSN 0001-0782. doi: 10.1145/3191513. URL https://doi.org/10.1145/3191513.

Eduardo Mosqueira-Rey, Elena Hernández-Pereira, David Alonso-Ríos, José Bobes-Bascarán, and Ángel Fernández-Leal. Human-in-the-loop machine learning: a state of the art. *Artificial Intelligence Review*, 56(4):3005–3054, Apr 2023. ISSN 1573-7462. doi: 10.1007/s10462-022-10246-w. URL https://doi.org/10.1007/s10462-022-10246-w.

Anis Najar and Mohamed Chetouani. Reinforcement learning with human advice: A survey. *Frontiers in Robotics and AI*, 8, 2021. ISSN 2296-9144. doi: 10.3389/frobt.2021.584075. URL https://www.frontiersin.org/articles/10.3389/frobt.2021.584075.

Simon Ott, Christian Meilicke, and Matthias Samwald. SAFRAN: An interpretable, rule-based link prediction method outperforming embedding models. In *3rd Conference on Automated Knowledge Base Construction*, pp. 1–18, 2021. URL https://openreview.net/forum?id=jCt9S_3w_S9.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F. Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback. In *NeurIPS*, 2022. URL http://papers.nips.cc/paper_files/paper/2022/hash/b1efde53be364a73914f58805a001731-Abstract-Conference.html.

Omkar M Parkhi, Andrea Vedaldi, Andrew Zisserman, and C. V. Jawahar. Cats and dogs. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3498–3505, 2012. doi: 10.1109/CVPR.2012.6248092.

Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 779–788, 2016. doi: 10.1109/CVPR.2016.91.

Tim Rocktäschel and Sebastian Riedel. End-to-end differentiable proving. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 30, pp. 1–15. Curran Associates, Inc., 2017.

Ali Sadeghian, Mohammadreza Armandpour, Patrick Ding, and Daisy Zhe Wang. Drum: End-to-end differentiable rule mining on knowledge graphs. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems*, pp. 1–11, Red Hook, NY, USA, 2019. Curran Associates Inc. doi: 10.5555/3454287.3455662. URL https://dl.acm.org/doi/10.5555/3454287.3455662.

Simon Schramm and Ute Schmid. Inductive logic programming for explainable graph clustering. In *2023 IEEE International Conference on Knowledge Graph (ICKG)*, pp. 235–242, 2023. doi: 10.1109/ICKG59574.2023.00034.

Simon Schramm, Christoph Wehner, and Ute Schmid. Comprehensible artificial intelligence on knowledge graphs: A survey. *Journal of Web Semantics*, 79:100806, 2023. ISSN 1570-8268. doi: https://doi.org/10.1016/j.websem.2023.100806. URL https://www.sciencedirect.com/science/article/pii/S1570826823000355.

Anna Seidel and Lothar von Falkenhausen. The emperor and his councillor laozi and han dynasty taoism. *Cahiers d'Extrême-Asie*, 17:125–165, 2008. ISSN 07661177, 21176272. URL http://www.jstor.org/stable/44171473.

Damian Szklarczyk, Annika L Gable, Katerina C Nastou, David Lyon, Rebecca Kirsch, Sampo Pyysalo, Nadezhda T Doncheva, Marc Legeay, Tao Fang, Peer Bork, Lars J Jensen, and Christian von Mering. The STRING database in 2021: customizable protein–protein networks, and functional characterization of user-uploaded gene/measurement sets. *Nucleic Acids Research*, 49(D1):D605–D612, 11 2020. ISSN 0305-1048. doi: 10.1093/nar/gkaa1074. URL https://doi.org/10.1093/nar/gkaa1074.

Stefano Teso, Öznur Alkan, Wolfgang Stammer, and Elizabeth Daly. Leveraging explanations in interactive machine learning: An overview. *Frontiers in Artificial Intelligence*, Volume 6 - 2023, 2023.

ISSN 2624-8212. doi: 10.3389/frai.2023.1066049. URL https://www.frontiersin.org/journals/artificial-intelligence/articles/10.3389/frai.2023.1066049.

Guojia Wan, Shirui Pan, Chen Gong, Chuan Zhou, and Gholamreza Haffari. Reasoning like human: Hierarchical reinforcement learning for knowledge graph reasoning. In Christian Bessiere (ed.), *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20*, pp. 1926–1932. International Joint Conferences on Artificial Intelligence Organization, 7 2020. doi: 10.24963/ijcai.2020/267. URL https://doi.org/10.24963/ijcai.2020/267. Main track.

Christoph Wehner, Francis Powlesland, Bashar Altakrouri, and Ute Schmid. Explainable online lane change predictions on a digital twin with a layer normalized lstm and layer-wise relevance propagation. In Hamido Fujita, Philippe Fournier-Viger, Moonis Ali, and Yinglin Wang (eds.), *Advances and Trends in Artificial Intelligence. Theory and Practices in Artificial Intelligence*, pp. 621–632, Cham, 2022. Springer International Publishing. ISBN 978-3-031-08530-7.

Christoph Wehner, Maximilian Kertel, and Judith Wewerka. Interactive and intelligent root cause analysis in manufacturing with causal bayesian networks and knowledge graphs. In *2023 IEEE 97th Vehicular Technology Conference (VTC2023-Spring)*, pp. 1–7, 2023. doi: 10.1109/VTC2023-Spring57618.2023.10199563.

Christoph Wehner, Chrysa Iliopoulou, Ute Schmid, and Tarek R. Besold. From latent to lucid: Transforming knowledge graph embeddings into interpretable structures with kgeprisma, 2025. URL https://arxiv.org/abs/2406.01759.

Norbert Wiener. Some moral and technical consequences of automation. *Science*, 131(3410):1355–1358, 1960. doi: 10.1126/science.131.3410.1355. URL https://www.science.org/doi/abs/10.1126/science.131.3410.1355.

Ronald J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8(3):229–256, May 1992. ISSN 1573-0565. doi: 10.1007/BF00992696. URL https://doi.org/10.1007/BF00992696.

Bishan Yang, Wen-tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. Embedding entities and relations for learning and inference in knowledge bases. In Yoshua Bengio and Yann LeCun (eds.), *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. URL http://arxiv.org/abs/1412.6575.

Fan Yang, Zhilin Yang, and William W Cohen. Differentiable learning of logical rules for knowledge base reasoning. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 30, pp. 1–10. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/0e55666a4ad822e0e34299df3591d979-Paper.pdf.

Denghui Zhang, Zixuan Yuan, Hao Liu, Xiaodong lin, and Hui Xiong. Learning to walk with dual agents for knowledge graph reasoning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(5): 5932–5941, Jun. 2022. doi: 10.1609/aaai.v36i5.20538. URL https://ojs.aaai.org/index.php/AAAI/article/view/20538.

Jing Zhang, Bo Chen, Lingxi Zhang, Xirui Ke, and Haipeng Ding. Neural, symbolic and neural-symbolic reasoning on knowledge graphs. *AI Open*, 2:14–35, 2021. ISSN 2666-6510. doi: https://doi.org/10.1016/j.aiopen.2021.03.001. URL https://www.sciencedirect.com/science/article/pii/S2666651021000061.

A Glossary

Table 10: Glossary of Symbols

Symbol	Description
Knowledge Graph and Link Prediction Task	
$\mathcal{G}$	A knowledge graph
V	A set of vertices (i.e., entities/nodes)
v	A vertex (i.e., entity/node)
v_h^q	The head vertex of a query
v_t^q	The tail vertex of a query
v_c	Vertex at step c
E	A set of edges (i.e., relations/links)
E_{valid}	A validation set with edges
e	An edge (i.e., relation/link)
e_{loop}	A specific type of edge that starts and ends at the same vertex
Σ_V	An alphabet with symbols for the vertices
σ_v	A vertex symbol
Σ_E	An alphabet with symbols for the edges
σ_e	An edge symbol
σ_e^q	The edge symbol of a missing relation in a query
h	The head mapping
t	The tail mapping
ℓ_V	The vertex language mapping
ℓ_E	The edge language mapping
P	Set of all possible tail link prediction mappings
p	Tail link prediction mapping
q	A query to predict a missing tail
Markov Decision Process	
MDP	Markov Decision Process
$\mathcal{S}$	Set of states
s_c	State at step c
c	Step number
C	Final step number
$\mathcal{A}$	Set of actions
$\mathcal{A}_{c_s}$	A set of actions available in the state at step c
a	An action
a_c	Selected action at step c
δ	Transition mapping
$\mathcal{R}$	Reward function
H_r	A set of latent reward histories
h_r	A latent reward history
Policy Network	
π	Policy
d_c	Probability distribution over all available actions at step c
h_c^d	A latent probability distribution history
$lnLSTN$	A layer normalized long short term memory layer
$softmax$	A softmax function
$leakyReLU$	A leaky ReLU function
α_{c-1}	Embedding of the action at step $c-1$
ς_c	Embedding of the state at step c
A_c	Stacked embedding of all available actions at step c
W	Weights vector of a dense layer
B	Bias vector of a dense layer
w	A walk through of the agent (i.e., path)
$w_{\sigma_e}^1$	A walk for a query with σ_e
κ_c	One reasoning step at a time step c
Reward Ensemble	
$\hat{r}_c$	Mean over all rewards at step c
r_c^i	Reward of ensemble member i at step c
i	Index of the reward ensemble member
Alignment	
Λ	Human oracle
$\omega_{\sigma_e}^1$	Embedded reasoning paths $w_{\sigma_e}^1$ for a query with σ_e
D	Database for storing preferences over walks w
μ	Preference factor