GaussianPath:A Bayesian Multi-Hop Reasoning Framework for Knowledge Graph Reasoning

Guojia Wan[1,2,3], **Bo Du**[1,2,3*]

[1]National Engineering Research Center for MultimediaSoftware, Wuhan University, China.
[2]Institute of Artificial Intelligence, School of Computer Science, Wuhan University, China.
[3]Hubei Key Laboratory of Multimedia and Network Communication Engineering, Wuhan University, China.
guojiawan@whu.edu.cn, dubo@whu.edu.cn

Abstract

Recently, multi-hop reasoning over incomplete Knowledge Graphs (KGs) has attracted wide attention due to its desirable interpretability for downstream tasks, such as question answer and knowledge graph completion. Multi-Hop reasoning is a typical sequential decision problem, which can be formulated as a Markov decision process (MDP). Subsequently, some reinforcement learning (RL) based approaches are proposed and proven effective to train an agent for reasoning paths sequentially until reaching the target answer. However, these approaches assume that an entity/relation representation follows a one-point distribution. In fact, different entities and relations may contain different certainties. On the other hand, since REINFORCE used for updating the policy in these approaches is a biased policy gradients method, the agent is prone to be stuck in high reward paths rather than broad reasoning paths, which leads to premature and suboptimal exploitation. In this paper, we consider a Bayesian reinforcement learning paradigm to harness uncertainty into multi-hop reasoning. By incorporating uncertainty into the representation layer, the agent trained by RL has uncertainty in a region of the state space then it should be more efficient in exploring unknown or less known part of the KG. In our approach, we build a Bayesian Q-learning architecture as a state-action value function for estimating the expected long-term reward. As initialized by Gaussian prior or pre-trained prior distribution, the representation layer drives uncertainty that allows regularizing the training. We conducted extensive experiments on multiple KGs. Experimental results show a superior performance than other baselines, especially significant improvements on the automated extracted KG.

1 Introduction

Knowledge graphs (KGs) such as WordNet (Bollacker et al. 2008), Yago (Suchanek, Kasneci, and Weikum 2007), NELL (Mitchell et al. 2018) contain numerous well-structured facts as triplets, e.g. $(Trump, isPresident, USA)$, to support a variety of downstream AI-applications such as question answer (QA), semantic search, and recommendation system. Knowledge graph reasoning (KGR) is a fundamental task to fulfill answering complex queries on large-scale incomplete knowledge graphs (Nickel et al. 2015; Ji et al. 2020). More specifically, the reasoning agent is expected to infer missing knowledge based on observed knowledge, where is usually modeled as a link prediction problem to predict potential candidate entities/relations for a query $(e_s, r, ?)/(e_s, ?, e_t)$.

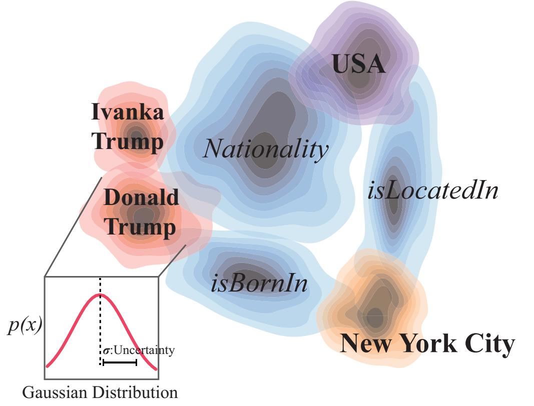

Figure 1: An illustrated toy example which generated from randomized two-dimensional Gaussian distribution for representing entities and relations in a partial KG. The mean of an entity indicates its position, and the variance indicates uncertainty.

The past years have seen the rapid development of knowledge graph embedding (KGE) in numerous studies (Bordes et al. 2013; Wang et al. 2017) focusing on automated reasoning on KGs, where these approaches embed both entities and relations into a continuous low-dimensional vector space. However, these approaches only learn single-step reasoning. To learn chains of reasoning paths over a KG, many reinforcement learning (RL) based methods (Xiong, Hoang, and Wang 2017; Das et al. 2018; Shen et al. 2018; Lin, Socher, and Xiong 2018) have been proposed to learn a query inferring agent via effective path searching. These methods have proven powerfully, and exhibit a desirable property that provides interpretable reasoning paths for queries. For example, give a query $(Trump, isPresident, ?)$, not only these models output a confidence score, but also provide a series of paths to explain why $isPresident(Trump, USA)$ holds, such as $Trump \xrightarrow{isCitizenOf} USA$, $Trump \xrightarrow{WorkAt} WhiteHouse \xrightarrow{LocatedIn} USA$.

[*]Corresponding author.

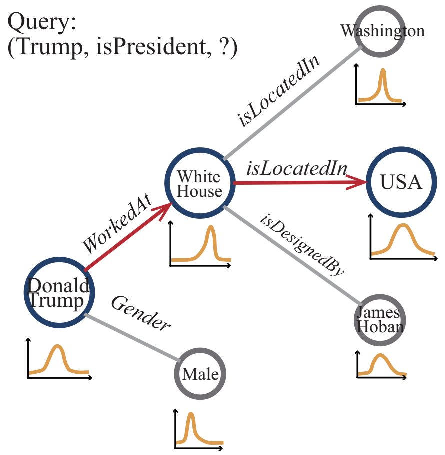

Figure 2: An example of Knowledge Graph Reasoning based on GaussianPath.

However, there are two limitations of existing RL-based multi-hop reasoning approaches (Xiong, Hoang, and Wang 2017; Das et al. 2018; Shen et al. 2018). Firstly, these approaches align each entity/relation a one-point distribution, i.e. a finite real-valued vector, whereas different entities and relations may contain different certainties. As Figure 1 shown, the uncertainty of relation *Nationality* is larger than *isBornIn* and *isLocatedIn* when inferring a query $(Donald\_Trump, isCitizenOf, ?)$. Accordingly, one-point distribution does not naturally express the uncertainty of a concept semantic in a KG. Secondly, existing methods employ the strategy of random sampling (Das et al. 2018) or greedy (Xiong, Hoang, and Wang 2017; Shen et al. 2018; Lin, Socher, and Xiong 2018) that results in a low probability of reaching positive rewards.

An alternative strategy is Bayesian approach, which takes into account the underlying uncertainty, and has proven effective in many traditional gaming tasks (Derman et al. 2020; Jeong and Lee 2018). The major incentives for incorporating Bayesian reasoning in RL are: 1) it provides an elegant approach to action-selection as a function of the uncertainty in learning, which is a principled way to tackle the exploration-exploitation problem; 2) it implicitly facilitates regularization. By assuming a prior on weights, we mitigate the trap of letting a few reasoning paths steer the agent away from the true parameters; and 3) it enables to incorporating prior knowledge into Markov decision process (MDP).

In this paper, we advocate moving beyond one-point distribution estimate to Bayesian inference for modeling uncertainty of multi-hop reasoning. Firstly, each entity/relation is aligned with a multi-dimensional Gaussian distribution $(\mathcal{N}(\mu, \Sigma))$. Therefore the state or the action at each time t also follows a joint distribution dependent on the original distributions. Secondly, we approximate the Q-function by Bayesian neural network architecture, where Bayesian LSTM (Fortunato, Blundell, and Vinyals 2017) encodes the state and Bayesian linear regression as an output layer predicts the expected long-term rewards for the possible actions. Based on the Bayesian approach, Thompson sampling (Thompson 1933) is used to pick an action that trades-off the exploitation-exploration dilemma. Finally, by minimizing Kullback-Leibler (KL) divergence with the true Bayesian posterior of the reasoning paths, the model learns a variational approximation to the Bayesian posterior distribution on the uncertainty of entity/relation representation. We employ unbiased Monte Carlo estimates of the gradients, Bayes by Backprop (Blundell et al. 2015) to optimize the posterior weights effectively.

Experimental results present that incorporating uncertainty into RL-based multi-hop reasoning yields better performance than one-point distribution representation, especially shows a significant improvement in noisy KGs. Additionally, we find that pre-trained Gaussian embedding as the prior distribution of each entity/relation can accelerate the training process. We conducted both knowledge graph completion tasks on standard benchmarks. The experimental results demonstrate the effectiveness of our approach.

Our contributions are as follows:

- We propose a Bayesian multi-hop reasoning paradigm, GaussianPath, for knowledge graph reasoning, aiming to capture the uncertainty of a reasoning path, which has been rarely studied in existing RL-based approaches yet.
- We construct a trainable Bayesian neural network architecture to approximate Q-function, which allows learning uncertainty of concept semantics and dealing with the trade-off between exploration and exploitation.
- We conducted extensive experiments on existing benchmark KGs. The results show that our model achieves competitive performance.

2 Related Work

2.1 Knowledge Graph Reasoning

Automated reasoning on KGs has been being a challenging problem as well as a hot topic. Earlier proposed symbolic logical reasoning based on expert system (McCarthy 1960; Quinlan 1990) suffers from poor generalization performance and the curse of dimension, despite of its high accuracy. To address the problem, KGE has been proposed to associate entities and relations into low dimensional continuous vector spaces (Bordes et al. 2013; Yih et al. 2011; Nickel et al. 2015; Wang et al. 2017). Since then, embedded spaces based on various geometry properties has been extensively studied (Wang et al. 2014; Lin et al. 2015; Trouillon et al. 2017; Xiao, Huang, and Zhu 2016a; Sun et al. 2018). Particularly, KG2E (He et al. 2015) has first investigated the uncertainty of KGE, and propose Gaussian embedding. Furthermore, TransG (Xiao, Huang, and Zhu 2016b) demonstrates the kind of uncertainty helps to handle multiple semantic issue. Other works (Chen et al. 2019; Ding et al. 2018) have studied the uncertainty from different perspectives. However, as single-hop reasoning methods, these approaches are limited in dealing with multi-hop reasoning scenarios, such as QA (Zhang et al. 2018).

2.2 Multi-Hop Reasoning

From its earlier on, PRA (Lao, Mitchell, and Cohen 2011) builds a linear regression to aggregate discrete path features which are extracted from a KG via random walk. Unfortunately, random walk is still computationally expensive due to traversing the entire graph. Neelakantan et al. (Neelakantan, Roth, and McCallum 2015) leverage a recurrent neural network (RNN) to compose the implications of a reasoning chain about conjunctions of multi-hop relations. NeuralLP (Yang, Yang, and Cohen 2017) proposes an end-to-end differentiable logical rules learning system for knowledge graph reasoning, but limited in expressing complex rules nevertheless. DIVA (Chen et al. 2018) situates multi-hop reasoning in the context of variational inference in latent variable probabilistic graphical models.

Recently, reinforcement learning has shown promising potential to model reasoning systems on a KG owing to its flexibility and interpretability (Stoica et al. 2020; Li and Cheng 2019; Bansal et al. 2019; Lv et al. 2019). DeepPath (Xiong, Hoang, and Wang 2017) is the first RL-based multi-hop reasoning approach for knowledge graph reasoning. For dealing with its limitations of which is only applicable for queries $(e_s, ?, e_t)$, Das et al. (Das et al. 2018) propose MINERVA to extend DeepPath by encoding a state chain into LSTM architecture. Afterwards, Lin et al. (Lin, Socher, and Xiong 2018) propose a reward shaping method to reduce the impact of false supervision for RL-based multi-hop reasoning methods. Additionally, it uses the trick termed as action dropout technique so as to introduce randomness into path search. More recently, Wan et al. (Wan et al. 2020) propose a hierarchical reinforcement learning framework to perform multi-hop reasoning. However, all of these approaches are built on a policy gradients method (REINFORCE (Williams 1992)) which usually has a large variance and heavily depends on an initial policy. Subsequently, Shen et al. (Shen et al. 2018) propose M-walk to adopt Monte Carlo Tree Search (MCTS) on Q-learning to deal with the issue. Nevertheless, there are rare works incorporating uncertainty into multi-hop reasoning methods.

3 Preliminaries and Notations

The notation table is shown in Table 1. Several key definitions are given as follows.

Definition 1 (Knowledge Graph)**.** *A Knowledge Graph is a directed graph $G = (\mathcal{E}, \mathcal{R}, U)$, where $\mathcal{E}$ is a set of entities, $\mathcal{R}$ is a set of relations, and U is a set of edges. $e \in \mathcal{E}$ is an entity. $r \in \mathcal{R}$ is a relation. $u \in U$ is an edge (e_s, r, e_t) that points from the source entity e_s to the target entity e_t.*

Definition 2 (Knowledge Graph Reasoning)**.** *Given a query among three cases $(e_s, r, ?)$, $(?, r, e_t)$, $(e_s, ?, e_t)$, Knowledge Graph Reasoning aims to predict the missing element of ? through a k-hop reasoning path $e_1 \xrightarrow{r_1} e_2 \xrightarrow{r_2} \cdots \xrightarrow{r_k} e_{k+1}$.*

Example: Given (Trump, isPresident,?), a possible 2-hop reasoning path is $Trump \overset{WorkAt}{\longrightarrow} WhiteHouse \overset{LocatedIn}{\longrightarrow} USA$.

Symbol	Meaning	Symbol	Meaning
$\mathcal{E}$	Entity set	e	Entity
$\mathcal{R}$	Relation set	r	Relation
e_s	Source entity	e_t	Target entity
G	KG	U	Edge set
S	State set	s	State
$\mathcal{A}$	Action set	a	Action
$R(\cdot)$	Reward function	γ	Discount factor
π	Policy	θ	Parameters
τ	Trajectory	$\mathcal{D}$	Training samples
ϕ	Bayesian regression	$f_x(\cdot)$	Distribution of x
Q	State-value function	;	Concatenate

Table 1: Annotation table

4 Methodology

4.1 Knowledge Graph Reasoning as a Markov Decision Process

We formulate knowledge graph reasoning as a MDP, which is described as a tuple $(\mathcal{S}, \mathcal{A}, \mathcal{T}, R)$. Each elements is elaborated below.

States The state s_i at step i is defined as a tuple (e_i, e_s, o), where $e_i \in \mathcal{E}$ is the current entity, e_s is the source entity. o denotes the query objective. Concretely, o is e_t under the task $(e_s, ?, e_t)$, otherwise r under the task $(e_s, r, ?)$. Given a query pair (e_s, o), the starting state is represented as (e_s, e_s, o). The final state is (e_t, e_s, o) if reaching the target entity otherwise $('STOP', e_s, o)$ within the max length T. After taking action, the agent will move to the next state.

Actions The action space $\mathcal{A}_{s_i}$ for the state $s_i = (e_i, e_s, o)$ is the set of outgoing edges of the current entity e_i, $\mathcal{A}_{s_i} = \{(r, e) | (e_i, r, e) \in U, e \notin \{e_0, e_1, \cdots, e_i\}\}$, where we remove entities to guarantee reasoning paths acylic. Beginning with the source entity e_s, the agent uses the Q-function to predict the most promising path, and it then extends its path at each step until it reaches the target entity e_t.

Transition The transition $\mathcal{T}$ is the state transition probability used to identify the probability distribution of the next state, which is defined as $\mathcal{T}(s_{i+1}|s_i, a_i)$.

Reward For each step i within a trajectory τ, we set up the reward function as $R(s_i) = \mathbb{I}[s_{End} = (e_t, e_s, o)]$, where s_{End} is the terminal state of τ. To put it another way, if the agent reaches the correct objective entities, it will receive a reward of 1, otherwise 0.

4.2 Model Uncertainty of Multi-Hop Reasoning

As just mentioned, one-point distribution maybe insufficient to model the uncertainty of entity/relation KGs. Therefore, we propose to use a Gaussian distribution to represent an entity/relation,

$$\begin{aligned} \mathbf{e} &\sim \mathcal{N}(\mu_e, \mathbf{\Sigma}_e) \\ \mathbf{r} &\sim \mathcal{N}(\mu_r, \mathbf{\Sigma}_r), \end{aligned} \tag{1}$$

where $\mu_e, \mu_r \in \mathbb{R}^d$ are mean vectors, $\mathbf{\Sigma}_e, \mathbf{\Sigma}_r \in \mathbb{R}^{d \times d}$ are covariance matrices (currently with diagonal covariance for computing efficiency). Obviously, a state or an action defined in the above also follows a joint distribution function as

$$f_{\mathcal{S}}(\mathbf{s}_i) := \iiint f_{e_i}(e_i) f_{e_s}(e_s) f_o(o) de_i de_s do, \quad (2)$$

$$f_{\mathcal{A}}(\mathbf{a}) := \iint f_e(e) f_r(r) dedr. \quad (3)$$

As iteratively interacting with the environment, the agent following an unknown state-action distribution extends its path from source entity to target entity. With training the agent, the posterior of these Gaussian distributions will begin to converge, and uncertainty can decrease, and so the agent will become more deterministic. Accordingly, we can describe the plausibility of a reasoning path τ under policy π as

$$\mathcal{F}(\tau|\pi) = \mathbb{P}(s_0) \prod_{t=1}^{T} \mathcal{T}(s_{t+1}|s_t, a_t, \pi, \theta), \quad (4)$$

where τ obeys Assumption 1 and its length $\leqslant T$. $\mathcal{F}$ is a typical Bayesian inference over the Markov chain dependent on the distributions of $\forall e \in \mathcal{E}, r \in \mathcal{R}$. We can observe that the uncertainty of $\mathbf{e}, \mathbf{r}$ will be sequentially transmitted to $\mathcal{F}$, resulting in the uncertainty of predicting a reasoning path.

4.3 Bayesian State-action Value Function

We use reinforcement learning to train an agent that behaves like $\mathcal{F}$. For each observation, the agent is evaluated by a state-value function, i.e. Q-function. Let $Q_\pi(s_t, a)$ denote Q-function. Thus, the corresponding Bellman equation (O'Donoghue et al. 2018) under a policy π, starting off from state s_t and taking action a in the action space $\mathcal{A}_{s_{t+1}}$,

$$Q_\pi(s_t, a) = \mathbb{E}_\pi[R(s_t) + \gamma \mathbb{E}_\pi[Q(s_{t+1}, a_{t+1})]]. \quad (5)$$

Given the current state and action, the environment stochastically proceed to a successor state s_{t+1} under probability $\mathcal{T}$ and provides a reward $R(s_t)$. For a given policy π and Markovian assumption of the model, we can rewrite the equation for the Q-functions as follows:

$$\begin{aligned} &Q_\pi(s_t, a_t) = R(s_t) + \\ &\gamma \sum_{s_{t+1}, a_{t+1}} \mathcal{T}(s_{t+1}|s_t, a_t) \pi(a_{t+1}|s_{t+1}) Q_\pi(s_{t+1}, a_{t+1}). \end{aligned} \quad (6)$$

Due to the large combination space of state-action pair (s, a) in a KG environment, it is hard to obtain Q-function directly from Eq. 6. For approximating the Q-function, we utilize the DQN (Mnih et al. 2013) architecture with modification built by Bayesian neural networks (Hernández-Lobato and Adams 2015), where the weights follow a prior distribution. Firstly, we encode the current state s_t into a latent vector $h^{(t)}$ via Bayesian LSTM (Fortunato, Blundell, and Vinyals 2017),

$$h^{(t)} = BayesianLSTM(s_t, h^{(t-1)}). \quad (7)$$

Bayesian LSTM is trainable for random variables, allowing the input of a probability distribution then outputting a probability distribution. Then we employ a Bayesian linear regression layer $\phi(\cdot)$ to learn the Q-value for each action,

$$Q(s_t, a) = \phi(h^{(t)})^\top w_a, \quad (8)$$

where $w_a = [r_{t+1}; e_{t+1}], \forall a \in \mathcal{A}_{s_t}$.

4.4 Off-Policy for Efficiency Exploration

Existing RL-based approaches (Xiong, Hoang, and Wang 2017; Das et al. 2018; Lin, Socher, and Xiong 2018) usually use REINFORCE (Williams 1992) to optimize objective function. However, REINFORCE encourages extending the next state with high reward, therefore biases the searching optimal policies as well as leads to unstable training with high variance (Guu et al. 2017; Agrawal and Goyal 2013). In order to deal with the problem, we apply an off-policy to find the optimal policy.

Our off-policy is divided into an optimization policy and an execution policy. The optimization policy is defined as

$$Q^*(s, a) = \max_a(Q(s, a)), \forall a \in \mathcal{A}_s. \quad (9)$$

As any greedy with respect to Q^* is optimal (Bellman 1958), our optimization policy is a greedy policy for guaranteeing exploitation. Then, the optimal state-action value function of an action may be written in terms of the optimal of its successor states as

$$\begin{aligned} &Q^*_\pi(s_t, a_t) = R(s_t) + \\ &\gamma \sum_{s_{t+1}, a_{t+1}} \mathcal{T}(s_{t+1}|s_t, a_t) Q^*_\pi(s_{t+1}, a_{t+1}), \end{aligned} \quad (10)$$

Notably, we do not know the transition kernel $\mathcal{T}$ in advance. As a consequence, we propose an execution policy to determine the next action towards the next state. At each step, we sample weights θ from the posterior distribution, then the agent acts the action with the max rewards,

$$a^* = \arg\max_a([Q_{\theta \sim p(\theta|\mathcal{D})}(s, a)]), \forall a \in \mathcal{A}_s. \quad (11)$$

Eq. 11 is in fact equivalent to Thompson sampling (Thompson 1933), which guarantees uncertainty through posterior sampling, and allows the agent with high uncertainty to explore effectively.

4.5 Optimization and Training

Bayes Variational Learning as Training Objective For estimating the posterior distribution $P(\theta|\mathcal{D})$, we apply Bayesian variational learning to find θ of a distribution $q(\theta)$ that minimizes the Kullback-Leibler (KL) divergence with the true Bayesian posterior on the parameters:

$$\begin{aligned} L(\theta^*) &= \min_\theta KL(q(\theta)||P(\theta|\mathcal{D})) \\ &= \min_\theta \int q(\theta) \log \frac{q(\theta)}{P(\theta)P(\mathcal{D}|\theta)} d\theta \\ &= \min_\theta KL(q(\theta)||P(\theta)) - \mathbb{E}_{\theta \sim q(\theta)}[\log P(\mathcal{D}|\theta)], \end{aligned} \quad (12)$$

where $P(\theta)$ is the prior distribution of θ. Minimizing Eq. 12 known as variational free energy (Wainwright and Jordan 2008) is equivalent to maximizing the log-likelihood $P(\mathcal{D}|\theta)$ given training data $\mathcal{D}$ subject to a KL penalty that acts as a regularizer.

Because the prior distribution of weights possesses uncertainty and follows Gaussian distribution, $Q(s, a)$ can be

Datasets	$\|\mathcal{E}\|$	$\|\mathcal{R}\|$	Triplets	Relation Tasks
FB15K-237	14505	237	272115	20
NELL995	75942	200	154231	12
WN18RR	40903	11	141422	-
UMLS	135	49	141422	-
Kinship	1043	26	10686	-

Table 2: Datasets

decomposed into the sum of the mean value and a Gaussian noise,

$$y_t = \bar{Q}_\theta(s_t, a_t) + \epsilon, \tag{13}$$

$$\hat{y}_t = r + \gamma \bar{Q}^*_\theta(s_{t+1}, a_{t+1}) + \epsilon', \tag{14}$$

where ϵ, ϵ' are zero mean Gaussian noise. Let $P(y_t = \hat{y}_t|s_t, a_t, \theta)$ be the predictive distribution $P(\mathcal{D}|\theta)$. Accordingly, the objective at the step t is to be

$$L(\tau, \theta) = KL(q(\theta)||P(\theta)) - \mathbb{E}_{q(\theta)}[\log P(y_t = \hat{y}_t|s_t, a, \theta)]. \tag{15}$$

Unfortunately, exactly minimizing this objective naively is computationally extensive. As a consequence, we approximate the objective by Bayes by Backprop (Blundell et al. 2015; Fortunato, Blundell, and Vinyals 2017) that is designed to learn the probability distribution on the weights of a neural network,

$$\begin{aligned} L(\tau, \theta) \approx \frac{1}{NT} \sum_{i=1}^{N} \sum_{t=0}^{T} [\log q(\theta_i) - \log P(\theta_i) \\ - \log P_i(y_t = \hat{y}_t|s_t, a, \theta)], \end{aligned} \tag{16}$$

where i denotes the i-th sample. We can estimate $P_i(y_t = \hat{y}_t|s_t, a, \theta), \epsilon, \epsilon'$ and gradients of Eq. 16 by repeating Monte Carlo sample drawn from the variational posterior $q(\theta)$.

Training Details The pseudo code of our approach is shown in Algorithm 1. All Gaussian priors for entities and relations are first initialized randomly following a uniform distribution or a prior distribution pre-trained by probability model, such as KG2E (He et al. 2015). θ_W are the parameters of the network architecture. θ_W are the parameters of neural network architecture. $\theta_{\mathcal{E}+\mathcal{R}}$ denotes the embedding layer of entities and relations.

Since $\theta_{\mathcal{E}+\mathcal{R}}$ are different types of geometric objects, the means should not be allowed to grow too large. Therefore we apply the following hard constraint when we estimate $\theta_{\mathcal{E}+\mathcal{R}}$,

$$\forall l \in \mathcal{E} + \mathcal{R}, ||\mu_l||_2 \leqslant 1. \tag{17}$$

For ensuring that the covariance matrices positive definite as well as reasonably sized, we constraint the diagonal covariance matrices within the hypercube $[c_{min}, c_{max}]^d$,

$$\begin{aligned} \forall l \in \mathcal{E} + \mathcal{R},\ c_{max} > c_{min} > 0, \\ c_{min}\mathbf{I} \prec \mathbf{\Sigma_l} \prec c_{max}\mathbf{I}. \end{aligned} \tag{18}$$

We implemented BayesianLSTM via a public implementation [1], in which Bayes by Backprop (Blundell et al. 2015) algorithm is used to calculate gradients. We unroll reasoning paths with the max length T. The 'STOP' action as a placeholder will be padded at the end of a reasoning path if the reasoning procedure ends or reaches the max length limitation. For further reducing variance, the Q-function is trained on mini-batches with the batch size B. Our code is available at https://github.com/BromothymolBlue/Gaupa.

Algorithm 1 GaussianPath

```
Initialize parameters $\theta = \theta_{\mathcal{E}+\mathcal{R}} \cup \theta_W$:
  $\forall l \in \mathcal{E} + \mathcal{R}, \mu_l \leftarrow \text{Uniform}(\frac{-6}{\sqrt{d}}, \frac{6}{\sqrt{d}})$,
  $\Sigma_l \leftarrow \text{Uniform}(c_{min}, c_{max})$,
  $\forall w \in \theta_W, w \leftarrow$ Xavier(Glorot and Bengio 2010).
Initialize training queries $\forall (e_s, r, e_t) \in U$
repeat
  Sampling a query:$(e_s, r, e_t) \leftarrow \text{Sample}(U)$
  Initialize $s_0 \leftarrow (e_s, e_s, o)$
  for $t \leftarrow 1$ to $T$ do
    Observe actions: $\forall a \in \mathcal{A}_{s_t}$
    for $i \leftarrow 1$ to $N$ do
      Sample parameters $\theta_i$ from the variational posterior $q(\theta)$
      Calculate Q-values $Q_{\theta_i}(s_t, a)$ on $\mathcal{A}_{\mathbf{s}_t}$
      $Q^*_{\theta_i}(a_{t+1}, s_{t+1}) \leftarrow \max_a(Q_{\theta_i}(a, s_t))$
      Get reward $r$
      $y_t \leftarrow Q_{\theta_i}(s_t, a)$
      $\hat{y}_t \leftarrow r + \gamma Q^*_{\theta_i}(s_{t+1}, a_{t+1})$
      $\nabla L_i \leftarrow$ Bayes by Backprop$(L(y_t, \hat{y}_t, \theta_i))$
    $a_{t+1} \leftarrow$ Thompson sampling$(Q_{\theta \sim q(\theta)}(s_t, a))$
    Execute the action and move to the next state:
    $a_t \leftarrow a_{t+1}, s_t \leftarrow s_{t+1}$
  Update parameters per $B$ episodes:
  $\theta \leftarrow \theta - \eta \sum_{t=1}^{T} \sum_{i=1}^{N} \nabla L_{i,t}$
until Converge $Q(s, a)$
Output: the policy, $\pi(s) = \arg\max_a(Q_{\theta \sim p(\theta|\mathcal{D})}(s, a))$
```

5 Experiments Settings

Datasets We performed experiments on five benchmark datasets: FB15K237(Dettmers et al. 2018), WN18RR(Dettmers et al. 2018), NELL995(Xiong, Hoang, and Wang 2017), UMLS(Das et al. 2018) and Kinship(Das et al. 2018). Details about these datasets are shown in Table 2.

Evaluation Protocols To evaluate the performance of KGR, we apply standard knowledge graph completion tasks following previous work (Das et al. 2018) on our approaches. We report Hit@1, 10 and mean reciprocal rank (MRR), and mean average precision (MAP) score for link prediction task.

Baselines In our experiments, we use following knowledge graph reasoning approaches for comparing: DistMult[3] (Yang et al. 2015), ConvE(Dettmers et al. 2018), NeuralLP(Yang, Yang, and Cohen 2017), MINERVA[2](Das et al. 2018), Multihop-KG(Lin, Socher, and Xiong 2018), M-walk

[1] https://github.com/piEsposito/blitz-bayesian-deep-learning

[2] https://github.com/shehzaadzd/MINERVA

Methods	FB15K237			WN18RR			NELL995			UMLS			Kinship		
	MRR	Hit@1	Hit@10	MRR	Hit@1	Hit@10	MRR	Hit@1	Hit@10	MRR	Hit@1	Hit@10	MRR	Hit@1	Hit@10
DistMult	0.417	0.324	0.600	0.462	0.431	0.524	0.641	0.552	0.783	0.868	0.821	0.967	0.614	0.487	0.904
ConvE	0.435	**0.341**	0.622	0.438	0.403	0.519	0.747	0.672	0.864	0.933	0.894	**0.992**	0.797	0.697	0.974
NeuraLP	0.227	0.166	0.348	**0.463**	0.376	0.657	-	-	-	0.778	0.643	0.962	0.619	0.475	0.912
MINERVA	0.293	0.217	0.456	0.448	0.413	0.513	0.725	0.663	0.831	0.825	0.728	0.968	0.720	0.605	0.924
Multihop-KG	0.407	0.327	0.564	0.450	0.418	0.517	0.727	0.656	0.844	**0.940**	**0.902**	0.992	0.865	0.789	0.982
M-walk	0.232	0.165	-	0.437	0.414	-	0.754	0.684	-	-	-	-	-	-	-
Ours	0.423	0.325	0.598	0.458	0.426	**0.663**	0.748	**0.691**	**0.894**	0.884	0.872	0.989	0.874	0.791	**0.987**
Ours(KG2E)	**0.440**	0.316	**0.638**	0.446	**0.437**	0.651	**0.756**	0.673	0.841	0.867	0.881	0.991	**0.884**	**0.812**	0.981

Table 3: Entity link prediction results on the lager KGs (FB15K237, WN18RR, NELL995) and the smaller KGs (UMLS and Kinship)

Tasks	**TE**	**PRA**	**DPath**	**MIN**	**Ours**
AthletePlaysForTeam	62.7	54.7	72.1	82.7	**85.7**
AthletePlaysInLeague	77.3	84.1	92.7	**95.2**	93.1
AthleteHomeStadium	71.8	85.9	84.6	92.8	**93.6**
AthletePlaysSport	87.6	47.4	91.7	**98.6**	96.7
TeamPlaySports	76.1	79.1	69.6	87.5	**90.3**
OrgHeadquaterCity	62.0	79.0	79.0	**94.5**	92.5
BornLocation	67.7	81.1	69.9	82.7	**88.1**
...					
Overall	72.3	71.8	78.41	88.4	**91.3**
contains	56.7	32.5	39.8	41.5	**68.4**
personNationality	44.2	42.1	52.8	**62.1**	61.9
musicianOrigin	38.2	18.5	23.7	23.8	**46.7**
adjoins	68.4	41.8	69.1	71.8	**79.1**
capitalOf	42.5	25.8	43.8	48.9	**52.3**
filmDirector	41.5	32.8	**45.6**	38.9	44.7
...					
Overall	45.3	31.5	39.8	42.3	**55.2**

Table 4: Relation link prediction results (MAP) with the MAP scores on NELL995 (Up) and FB15K-237 (Down).

(Shen et al. 2018), PRA (Lao, Mitchell, and Cohen 2011) and DeepPath (only for relation link prediction) (Xiong, Hoang, and Wang 2017).

Hyper-parameters Settings We set the dimension d of mean μ and Σ to 100. The layer number of BayesianLSTM is 1. The size of Hidden layer is 200. Learning rate η is 0.01. c_{min}/c_{max} is $0.01/0.4$. Other hyper-parameters settings are available in supplementary materials.

6 Results and Discussion

6.1 Link Prediction

Entity link prediction We conduct entity link prediction evaluation, which focuses on the effectiveness of our approach against the state-of-art knowledge graph reasoning approaches with respect to the predictive quality of finding ? for a query $(e_s, r, ?)$ or $(?, r, e_t)$. We mainly compare three sorts of methods: 1) single-step reasoning approaches (DistMult and ConvE); 2) logic rule learning (NeuralLP); 3) RL-based multi-hop reasoning (MINERVA, M-walk, Multihop-KG) on the smaller datasets (UMLS, Kinship) and the larger datasets (FB15K-237, WN18RR, NELL995). The baseline results are obtained from corresponding open resource implementations or reports. To further study the influence of prior distribution, we use pre-trained distribution of entity/relation KG2E (He et al. 2015) to initialize $\theta_{\mathcal{E}+\mathcal{R}}$.

On the smaller datasets (UMLS, Kinship), each of these KGs is with around 100 entities. As a result, short-term relationship is the main component in the reasoning paths, in which embedding-based approaches perform better. When we set the max length $T = 2$, i.e. degenerates into single-hop reasoning, our approach also shows a competitive performance, demonstrating that our method is effective in capturing short-term relationship in a local partial KG.

On larger datasets (FB15K237, WN18RR and NELL995), our approach outperforms most of baselines, and achieves state-of-the-art results on NELL995. NELL995 is an inborn noisy KG that is automatically constructed from WEB resource (Mitchell et al. 2018), which preserves partial unreliable triplets. We observe that our approach can significantly suppress the agent from extending the reasoning paths to unreliable triplets. More specifically, for each step of action selection under Thompson sampling strategy, the agent tends to select the action with the maximal expectation over all possible actions rather than the action with the maximal reward, therefore benefits the improvements on NELL995.

Relation link prediction This task aims to evaluate the quality of predicting the relation r for a query pair (e_s, e_t), which is a binary classification problem. We conducted the task on NELL995 and FB15K237. Unlike DeepPath (Xiong, Hoang, and Wang 2017) and PRA (Lao, Mitchell, and Cohen 2011) gather path features and train a linear regression for each separative relation, we train one model for all relations. Specifically, we implement it by adding a reasoning rule that prohibits the agent from visiting the target entity directly through the query relation r. The agent reasons under different samples θ from the posterior distribution, and we can obtain a score as a score to indicate the plausibility of r existing. if the score is higher than a threshold δ, then the query will be classified as positive. Otherwise, it will be classified as negative. The results are shown in Table 4. Our approach achieves comparable performance in most of the query relations

6.2 Convergence Analysis

To study impact of each proposed enhancement on the training procedure, we report the convergence curves of success

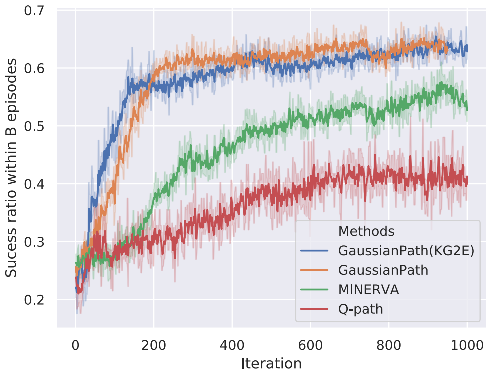

Figure 3: Convergence rate of reasoning success ration on NELL995. For fair comparison, we only include MINERVA.

ratio in Figure 3. For fair comparison, we fix the batch size $B = 512$. Success ratio denotes the proportion of reaching the correct entities in the candidate answer sets. From Figure 3, we can observe the convergence rate: GaussianPath(KG2E) $>$ GaussianPath $\gg$ MINERVA $>$ Q-path (seen in Section 6.3). The worst performance of Q-path is because inconsistency of off-policy leads to bias estimate for Q-function. MINERVA also has the problem due to high variance in estimating the policy gradients. After introducing entity/relation representation with uncertainty into Q-path, the curves exhibits faster convergence, demonstrating that more variability benefits handling the trade-off between exploration-exploitation. More specifically, the pretrained prior distribution KG2E slightly accelerates the convergence, indicating that GaussianPath succeeds in incorporating prior information into multi-hop reasoning.

6.3 Ablation Analysis

To understand the contributions of the different components, we ablate two components: 1) W/O Bayesian approach, named as Q-path, which degrades into DQN architecture sharing other settings of GaussianPath; 2) W/O Thompson Sampling, which employs the same execution policy as the optimization policy. As shown in Table 5, we can observe that: 1) Bayesian approach enables to improve RL-based multi-hop reasoning significantly; 2) Thompson sampling benefits the effective exploration thus further improve our model.

Components	MRR		
	FB15K237	WN18RR	NELL995
GaussianPath	**0.423**	**0.458**	**0.748**
Q-path	0.247	0.336	0.645
W/O TS	0.338	0.435	0.693

Table 5: Ablation study via the entity link prediction task.

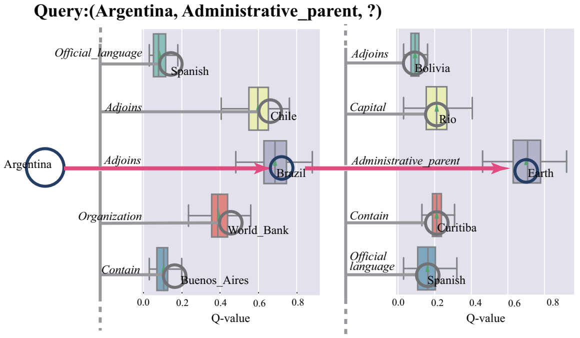

Figure 4: Box plots of two steps of the reasoning path.

6.4 Case Visualization of Reasoning Path

We provide a graphical visualization to better illustrate the mechanism of our approach through the reasoning path, $Argentina \xrightarrow{Adjoins} Brazil \xrightarrow{Administrative\_parent} Earth$. We construct box plots for every hop reasoning, where displays distribution in samples of a statistical population of the selected actions. In the hop-1 plot, we can observe that the expected Q-value of the actions, $(Organization, World/_Bank)$, $(Adjoins, Brazil)$ and $(Adjoins, Chile)$, are obviously larger than the rest actions. Note that the agent can visit the target entity $Earth$ through both $(Organization, World\_Bank)$ and $(Adjoins, Chile)$. Therefore the three deserve higher expected Q-values. Moreover, these actions still present a wide uncertainty, which means that the reasoning process is hard to be stuck in the local optimum. By contrast, The strategies used in existing methods (Das et al. 2018; Xiong, Hoang, and Wang 2017) is indifferent to the uncertainty of the actions and the expected rewards of sub-greedy ones, which employ uniform sampling or greedy (Shen et al. 2018) over the output of point values for the next movement. In the hop-2 plot, Similarly, we can observe that the target answer has a higher expected Q-value with an uncertainty, indicating the existence of a clear decision boundary between positive answers and negative answers.

7 Conclusions

We propose GaussianPath, a Bayes based RL multi-hop reasoning framework which expresses uncertainty of reasoning path. More specifically, we introduce Gaussian prior distribution to be entity/relation embeddings. This allows us to represent entity/relation not only as densities over a latent space, but driving uncertainty into agent interaction. In order to adapt the idea to multi-hop reasoning formulated as RL framework, we employ Bayesian neural network architecture to approximate Q-function. We propose an off-policy to balance the trade-off of exploration-exploitation dilemma. To learn this Bayesian posterior distribution of weights, we minimize the variational free energy on target Q-values. Experimental results show a comparable performance on KGC tasks. GaussianPath can leverage prior knowledge in the form of pre-trained Gaussian distribution, which slightly accelerates convergence on training.

Acknowledgements

This work was supported in part by National Key Research and Development Program of China under Grant: 2018AAA0101100, the National Natural Science Foundation of China under Grants 61822113, 41871243, the Natural Science Foundation of Hubei Province under Grants 2018CFA050 and the National Key R & D Program of China under Grant 2018YFA0605501, the Science and Technology Major Project of Hubei Province (Next-Generation AI Technologies) under Grant 2019AEA170. The numerical calculations in this paper have been done on the supercomputing system in the Supercomputing Center of Wuhan University.

References

Agrawal, S.; and Goyal, N. 2013. Thompson sampling for contextual bandits with linear payoffs. In *ICML*, 127–135.

Bansal, T.; Juan, D.-C.; Ravi, S.; and McCallum, A. 2019. A2N: attending to neighbors for knowledge graph inference. In *ACL*, 4387–4392.

Bellman, R. 1958. Dynamic programming and stochastic control processes. *Information and Control* 1(3): 228–239.

Blundell, C.; Cornebise, J.; Kavukcuoglu, K.; and Wierstra, D. 2015. Weight Uncertainty in Neural Network. In *ICML*, 1613–1622.

Bollacker, K.; Evans, C.; Paritosh, P.; Sturge, T.; and Taylor, J. 2008. Freebase: a collaboratively created graph database for structuring human knowledge. In *ACM SIGMOD International Conference on Management of Data*, 1247–1250.

Bordes, A.; Usunier, N.; Garcia-Duran, A.; Weston, J.; and Yakhnenko, O. 2013. Translating embeddings for modeling multi-relational data. In *NeurIPS*, 2787–2795.

Chen, W.; Xiong, W.; Yan, X.; and Wang, W. Y. 2018. Variational Knowledge Graph Reasoning. In *NAACL-HLT*, 1823–1832.

Chen, X.; Chen, M.; Shi, W.; Sun, Y.; and Zaniolo, C. 2019. Embedding uncertain knowledge graphs. In *AAAI*, 3363–3370.

Das, R.; Dhuliawala, S.; Zaheer, M.; Vilnis, L.; Durugkar, I.; Krishnamurthy, A.; Smola, A.; and McCallum, A. 2018. Go for a Walk and Arrive at the Answer: Reasoning Over Paths in Knowledge Bases using Reinforcement Learning. In *ICLR*.

Derman, E.; Mankowitz, D.; Mann, T.; and Mannor, S. 2020. A Bayesian Approach to Robust Reinforcement Learning. In *UAI*, 648–658.

Dettmers, T.; Minervini, P.; Stenetorp, P.; and Riedel, S. 2018. Convolutional 2D knowledge graph embeddings. In *AAAI*, 1811–1818.

Ding, B.; Wang, Q.; Wang, B.; and Guo, L. 2018. Improving Knowledge Graph Embedding Using Simple Constraints. In *ACL*, 110–121.

Fortunato, M.; Blundell, C.; and Vinyals, O. 2017. Bayesian recurrent neural networks. *arXiv preprint arXiv:1704.02798* .

Glorot, X.; and Bengio, Y. 2010. Understanding the difficulty of training deep feedforward neural networks. In *International Conference on Artificial Intelligence and Statistics*, 249–256.

Guu, K.; Pasupat, P.; Liu, E.; and Liang, P. 2017. From Language to Programs: Bridging Reinforcement Learning and Maximum Marginal Likelihood. In *ACL*, 1051–1062.

He, S.; Liu, K.; Ji, G.; and Zhao, J. 2015. Learning to represent knowledge graphs with gaussian embedding. In *CIKM*, 623–632.

Hernández-Lobato, J. M.; and Adams, R. 2015. Probabilistic backpropagation for scalable learning of bayesian neural networks. In *ICML*, 1861–1869.

Jeong, H.; and Lee, D. D. 2018. Bayesian Q-learning with Assumed Density Filtering. In *AAAI*.

Ji, S.; Pan, S.; Cambria, E.; Marttinen, P.; and Yu, P. S. 2020. A survey on knowledge graphs: Representation, acquisition and applications. *arXiv preprint arXiv:2002.00388* .

Lao, N.; Mitchell, T.; and Cohen, W. W. 2011. Random walk inference and learning in a large scale knowledge base. In *EMNLP*, 529–539.

Li, R.; and Cheng, X. 2019. DIVINE: A Generative Adversarial Imitation Learning Framework for Knowledge Graph Reasoning. In *EMNLP*, 2642–2651.

Lin, X. V.; Socher, R.; and Xiong, C. 2018. Multi-Hop Knowledge Graph Reasoning with Reward Shaping. In *EMNLP*, 3243–3253.

Lin, Y.; Liu, Z.; Sun, M.; Liu, Y.; and Zhu, X. 2015. Learning Entity and Relation Embeddings for Knowledge Graph Completion. In *AAAI*, 2181–2187.

Lv, X.; Gu, Y.; Han, X.; Hou, L.; Li, J.; and Liu, Z. 2019. Adapting Meta Knowledge Graph Information for Multi-Hop Reasoning over Few-Shot Relations. In *EMNLP*, 3367–3372.

McCarthy, J. 1960. *Programs with common sense*. RLE and MIT computation center.

Mitchell, T.; Cohen, W.; Hruschka, E.; Talukdar, P.; Yang, B.; Betteridge, J.; Carlson, A.; Dalvi, B.; Gardner, M.; Kisiel, B.; et al. 2018. Never-ending learning. *Communications of the ACM* 61(5): 103–115.

Mnih, V.; Kavukcuoglu, K.; Silver, D.; Graves, A.; Antonoglou, I.; Wierstra, D.; and Riedmiller, M. 2013. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602* .

Neelakantan, A.; Roth, B.; and McCallum, A. 2015. Compositional Vector Space Models for Knowledge Base Completion. In *ACL*, 156–166.

Nickel, M.; Murphy, K.; Tresp, V.; and Gabrilovich, E. 2015. A review of relational machine learning for knowledge graphs. *Proceedings of the IEEE* 104(1): 11–33.

O'Donoghue, B.; Osband, I.; Munos, R.; and Mnih, V. 2018. The uncertainty bellman equation and exploration. In *ICML*, 3836–3845.

Quinlan, J. R. 1990. Learning logical definitions from relations. *Machine Learning* 5(3): 239–266.

Shen, Y.; Chen, J.; Huang, P.-S.; Guo, Y.; and Gao, J. 2018. M-walk: Learning to walk over graphs using monte carlo tree search. In *NeurIPS*, 6786–6797.

Stoica, G.; Stretcu, O.; Platanios, E. A.; Mitchell, T. M.; and Póczos, B. 2020. Contextual Parameter Generation for Knowledge Graph Link Prediction. In *AAAI*, 3000–3008.

Suchanek, F. M.; Kasneci, G.; and Weikum, G. 2007. Yago: a core of semantic knowledge. In *WWW*, 697–706.

Sun, Z.; Deng, Z.-H.; Nie, J.-Y.; and Tang, J. 2018. RotatE: Knowledge Graph Embedding by Relational Rotation in Complex Space. In *ICLR*.

Thompson, W. R. 1933. On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika* 25(3/4): 285–294.

Trouillon, T.; Dance, C. R.; Gaussier, É.; Welbl, J.; Riedel, S.; and Bouchard, G. 2017. Knowledge graph completion via complex tensor factorization. *The Journal of Machine Learning Research* 18(1): 4735–4772.

Wainwright, M. J.; and Jordan, M. I. 2008. Graphical Models, Exponential Families, and Variational Inference. *Machine Learning* 1(1-2): 1–305.

Wan, G.; Pan, S.; Gong, C.; Zhou, C.; and Haffari, G. 2020. Reasoning Like Human: Hierarchical Reinforcement Learning for Knowledge Graph Reasoning. In *IJCAI*.

Wang, Q.; Mao, Z.; Wang, B.; and Guo, L. 2017. Knowledge graph embedding: A survey of approaches and applications. *IEEE Transactions on Knowledge and Data Engineering* 29(12): 2724–2743.

Wang, Z.; Zhang, J.; Feng, J.; and Chen, Z. 2014. Knowledge graph embedding by translating on hyperplanes. In *AAAI*, 2014.

Williams, R. J. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning* 8(3-4): 229–256.

Xiao, H.; Huang, M.; and Zhu, X. 2016a. From one point to a manifold: knowledge graph embedding for precise link prediction. In *IJCAI*, 1315–1321.

Xiao, H.; Huang, M.; and Zhu, X. 2016b. TransG: A Generative Model for Knowledge Graph Embedding. In *ACL*, 2316–2325.

Xiong, W.; Hoang, T.; and Wang, W. Y. 2017. DeepPath: A Reinforcement Learning Method for Knowledge Graph Reasoning. In *EMNLP*, 564–573.

Yang, B.; Yih, W.; He, X.; Gao, J.; and Deng, L. 2015. Embedding Entities and Relations for Learning and Inference in Knowledge Bases. In *ICLR*.

Yang, F.; Yang, Z.; and Cohen, W. W. 2017. Differentiable learning of logical rules for knowledge base reasoning. In *NeurIPS*, 2319–2328.

Yih, W.-t.; Toutanova, K.; Platt, J. C.; and Meek, C. 2011. Learning discriminative projections for text similarity measures. In *CoNLL*, 247–256.

Zhang, Y.; Dai, H.; Kozareva, Z.; Smola, A. J.; and Song, L. 2018. Variational Reasoning for Question Answering With Knowledge Graph. In *AAAI*.