Graph Stochastic Neural Process for Inductive Few-shot Knowledge Graph Completion

ZICHENG ZHAO*, Key Laboratory of Intelligent Perception and Systems for High-Dimensional Information, Ministry of Education, School of Computer Science and Engineering, Nanjing University of Science and Technology, China
LINHAO LUO*, Department of Data Science and AI, Monash University, Australia
SHIRUI PAN, School of Information and Communication Technology and Institute for Integrated and Intelligent Systems (IIIS), Griffith University, Australia
CHENGQI ZHANG, School of Computer Science and Technology, University of Technology Sydney, Australia
CHEN GONG†, Key Laboratory of Intelligent Perception and Systems for High-Dimensional Information, Ministry of Education, School of Computer Science and Engineering, Nanjing University of Science and Technology, China

Knowledge graphs (KGs) store enormous facts as relationships between entities. Due to the long-tailed distribution of relations and the incompleteness of KGs, there is growing interest in few-shot knowledge graph completion (FKGC). Existing FKGC methods often assume the existence of all entities in KGs, which may not be practical since new relations and entities can emerge over time. Therefore, we focus on a more challenging task called inductive few-shot knowledge graph completion (I-FKGC), where both relations and entities during the test phase are unknown before. Inspired by the idea of inductive reasoning, we cast I-FKGC as an inductive reasoning problem. Specifically, we propose a novel Graph Stochastic Neural Process approach (GS-NP), which consists of two major modules. In the first module, to obtain a generalized hypothesis (*e.g.*, shared subgraph), we present a neural process-based hypothesis extractor that models the joint distribution of hypothesis, from which we can sample a hypothesis for predictions. In the second module, based on the hypothesis, we propose a graph stochastic attention-based predictor to test if the triple in the query set aligns with the extracted hypothesis. Meanwhile, the predictor can generate an explanatory subgraph identified by the hypothesis. Finally, the training of these two modules is seamlessly combined into a unified objective function, of which the effectiveness is verified by theoretical analyses as well as empirical studies. Extensive experiments on three public datasets demonstrate that our method outperforms existing methods and derives new state-of-the-art performance.

*Both authors contributed equally to this research.
†Corresponding authors: Chen Gong.

Authors' Contact Information: Zicheng Zhao, Key Laboratory of Intelligent Perception and Systems for High-Dimensional Information, Ministry of Education, School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing, China, zicheng.zhao@njust.edu.cn; Linhao Luo, Department of Data Science and AI, Monash University, Melbourne, Australia, linhao.luo@monash.edu; Shirui Pan, School of Information and Communication Technology and Institute for Integrated and Intelligent Systems (IIIS), Griffith University, Queensland, Australia, s.pan@griffith.edu.au; Chengqi Zhang, School of Computer Science and Technology, University of Technology Sydney, Sydney, Australia, Chengqi.Zhang@uts.edu.au; Chen Gong, Key Laboratory of Intelligent Perception and Systems for High-Dimensional Information, Ministry of Education, School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing, China, chen.gong@njust.edu.cn.

ACM 2157-6912/2025/11-ART
https://doi.org/10.1145/3773908

CCS Concepts: • **Computing methodologies → Knowledge representation and reasoning**.

Additional Key Words and Phrases: Few-shot Learning, Knowledge Graph Completion, Neural Process, Inductive Reasoning, Graph Stochastic Attention

1 INTRODUCTION

Knowledge graphs (KGs) [20, 29], which contain factual knowledge in terms of structural triples, *i.e.*, (`head entity, relation, tail entity`), are widely utilized in various downstream applications, such as recommender systems [46, 48], community detection [14, 32], protein design [26], and web search [33]. However, KGs often suffer from incompleteness [13], limiting their practical applicability. Many approaches have been proposed to complete missing links by inferring from observed facts [3, 4, 43, 57]. However, these methods require sufficient training data for each relation, which is not always available. In other words, in real-world KGs, there is a significant amount of presented relations that are only associated with a couple of facts in KGs. This poses a great challenge for predicting missing facts within the KG. To tackle this issue, several few-shot knowledge graph completion (FKGC) methods [5, 36, 42, 62] have been proposed to predict missing facts for relations with limited supporting triples.

Existing few-shot knowledge graph completion (FKGC) methods typically follow the meta-learning framework [15, 16]. In this framework, the model is either fine-tuned [5, 42] on a *support set*, which contains a few triples related to a relation, or designs a matching network to calculate the similarity between *query set* (which consists of triples to be predicted) and *support set* for the few-shot task [50, 62, 65]. Despite the success, these methods often suffer from out-of-distribution (OOD) [17] and overfitting problems [8], which cannot extract a general pattern to derive new knowledge effectively. Moreover, they do not consider the dynamic nature of KGs, where new entities emerge with new relations simultaneously. As shown in Figure 1(a), relations and entities involved in the testing phase are all unseen to the original background knowledge graph (BG Graph). In light of their limitations, we focus on a more challenging setting, *i.e.,* **inductive few-shot knowledge graph completion (I-FKGC)**, where the model is required to predict missing facts for unseen relations and entities in the query set with limited supporting triples in the support set [17, 31].

The objective of I-FKGC is to incorporate new relations and new entities into KGs, which requires the model to exhibit strong inductive reasoning ability, enabling it to generalize effectively in unseen scenarios. Inspired by inductive reasoning [18], in this paper, we cast I-FKGC as an inductive reasoning problem. We attempt to find a hypothesis (*e.g.*, a pattern or a subgraph) shared among all triples in the support set and then test whether each triple in the query set aligns with this hypothesis for prediction. More specifically, we try to extract a shared hypothesis, as depicted in Figure 1(b), by examining the subgraphs surrounding the triples in the support set. By analyzing these subgraphs, we can observe that they exhibit similar structures, such as (`bake, requires, chef`), (`chef, works in, kitchen`) and (`play, requires, musician`), (`musician, works in, music room`). Consequently, we can extract a hypothesis: ρ: {(`head`, h_q, `requires, tail`, e_q), (`head`, e_q, `works in, tail`, t_q)}, which explains how to establish the relation r_q between the head entities and tail entities. For instance, bake requires the chef and the chef works in the kitchen. Based on this hypothesis, we can infer that it is highly likely for (`judge,` r_q`, courtroom`) from the query set to be true since it aligns with our hypothesis, *i.e.,* judge requires magistrate and magistrate works in courtroom.

Based on the above consideration, in this paper, we propose a novel Graph Stochastic Neural Process approach (`GS-NP`), which contains two major modules to effectively extract and apply the hypothesis for addressing the I-FKGC task. Existing methods extract the hypothesis by identifying the most commonly shared structure among the support set [17]. However, due to the limited number of support triples, the extracted hypothesis may not generalize effectively to the query set. Neural process (NP) [12] offers a new way to deal with limited data by modeling a distribution over the prediction function. Motivated by this, we propose a *neural process-based*

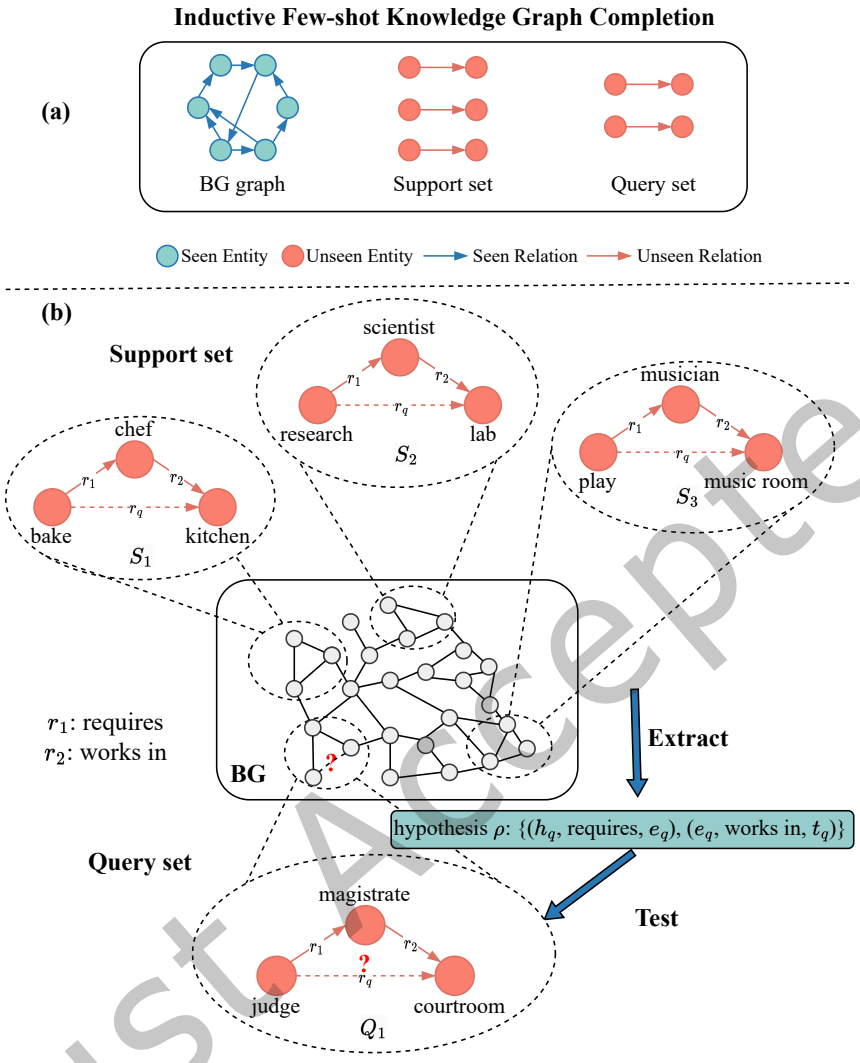

Fig. 1. (a) An illustration of inductive few-shot link prediction (I-FKGC) where new relations emerge with unseen entities simultaneously. (b) An example of the motivation behind our method where we try to test whether the query set aligns with the hypothesis extracted from the support set.

hypothesis extractor that treats the hypothesis as a latent variable and models its joint distribution instead of directly extracting it. This approach helps to alleviate overfitting and out-of-distribution issues. By sampling from this distribution, we can obtain the hypothesis for predictions. Then, we propose a *graph stochastic attention-based predictor* to test whether the query set aligns with the extracted hypothesis. Since the sampled hypothesis is a latent variable, we employ the stochastic attention mechanism [49] that transforms the hypothesis into a soft edge mask and identifies a subgraph used for predictions. In this way, our method can generalize well to the

query set under few-shot scenarios and provide explanations for predictions. The training of these two modules is seamlessly combined into a unified objective function, of which the effectiveness is verified by both theoretical analyses and empirical studies.

The main contributions of our paper can be summarized as follows:

- We propose a novel neural process approach for inductive few-shot knowledge graph completion. To the best of our knowledge, this is the first work to develop a neural process framework for I-FKGC.
- We propose a graph stochastic attention-based predictor to test whether the query set shares the same hypothesis extracted from the support set. Based on the extracted hypothesis, our method can generalize well to the query set under few-shot scenarios and provide explanations for predictions.
- We conduct extensive experiments on three typical datasets. Experimental results show that our model (GS-NP) outperforms existing methods, verifying the effectiveness of our method.

The structure of the paper is organized as follows: Section 2 reviews representative prior works on few-shot knowledge graph completion along with two key techniques integral to our methods. Section 3 provides a brief introduction to the concepts of neural processes and graph stochastic attention, and outlines the settings and common notations used throughout the paper. Section 4 elaborates on the motivation behind our research and details our proposed framework. Section 5 discusses the experimental evaluations and case studies conducted to validate our methods. Section 6 concludes the paper and discusses potential future research directions.

2 RELATED WORK

In this section, we review some representative prior works and two key techniques related to this paper.

2.1 Few-shot Knowledge Graph Completion

Existing few-shot knowledge graph completion (FKGC) methods can be roughly divided into two groups: meta-optimization-based and metric learning-based models. Meta-optimization-based methods attempt to quickly update the meta parameters via gradient descent on the few-shot data, making models seamlessly generalize to new relations. For instance, MetaR [5] proposes a relation-meta learner to extract relation-specific meta-information and then applies it to few-shot relational predictions. GANA [42] introduces a gated and attentive neighbor aggregator to improve the quality of generated embeddings and the relation meta representation. However, meta-optimization-based methods are often sensitive to the quality of given few-shot data and suffer from out-of-distribution problems [30]. Metric learning-based methods try to develop a matching network, which first encodes the triples in the support set and then measures the similarity between the query set and the support set. GMatching [62] proposes a neighbor encoder and LSTM matching network to measure the similarity, which is the first work to address the FKGC problem. FSRL [65] builds on the neighbor information aggregation mechanism of GMatching by introducing a fixed attention mechanism to consider multiple support triples. FAAN [50] extends FSRL by presenting a relation-specific adaptive neighbor encoder.

To handle unseen entities, the problem of inductive link prediction has also received much attention [7, 53, 59, 64, 66]. GraIL [53] utilizes the enclosing subgraph around the target triple for inductive link prediction. PatchCon [59] introduces a novel relation message-passing mechanism to capture the inductive features. Recently, there have been several attempts to develop foundation models for knowledge graph completion [25], particularly ULTRA [10], which is trained on massive datasets and demonstrates strong generalization capabilities. However, these methods focus on the conventional inductive setting and are not well-suited to the few-shot setting considered in this paper. To address this issue, CSR [17] combines the merit of existing FKGC and inductive link prediction methods, introducing a novel subgraph-based matching network. It tries to extract connection subgraphs among the support triples and tests whether the query set shares the same subgraph. Therefore, CSR is the only state-of-the-art method designed to address the I-FKGC task. However, due to the limited number of

supporting triples, the extracted subgraph may not generalize well to the query set, which limits its performance [10].

2.2 Neural Process Family

Neural Process (NP) [12] attempts to define a distribution over prediction functions with limited observed data, which can quickly be adapted to new tasks. CNP [11] encodes the observed data into a deterministic hidden variable, which fails to account for uncertainties. After that, there are several studies proposed to improve the neural process in various aspects. For instance, ANP [21] introduces the self-attention mechanism to better capture dependencies and model the distribution. SNP [51] is designed for sequential data and employs a recurrent neural network (RNN) to capture temporal correlation for better generalization. NP has been applied to many applications, such as recommender systems [28] and link prediction [27, 34]. Recently, NP-FKGC [36] proposes a normalizing flow-based NP for few-shot knowledge graph completion, and RawNP [66] proposes a relational anonymous walk-based NP to inductively predict the missing facts for unseen entities.

2.3 Stochastic Attention Mechanism

The stochastic attention mechanism is a novel attention mechanism based on a stochastic model that attempts to capture complicated dependencies and regularize the weights based on the input data. Early works [2, 49] adopt a normal distribution as the posterior distribution of the attention weight, satisfying the simplex constraints, *i.e.,* sum to one [2]. However, these methods cannot utilize back-propagation to optimize the attention weight. Recently, the Bayesian attention module [9] proposes a differentiable stochastic attention, which can be optimized during training. The stochastic attention has been applied in many applications. Xu et al. [63] adopt stochastic attention to capture the important regions in the image for image captioning. GSAT [40] extracts label-relevant subgraphs with graph stochastic attention to provide interpretable explanations for predictions.

3 PRELIMINARIES

In this section, we give a brief introduction to the key concepts in our paper and a formal definition of our problem. Commonly used notations are present in Table 1.

3.1 Neural Process

Combining the benefits of the stochastic process and neural networks, neural process (NP) [12] attempts to model the joint distribution over the prediction function $f : X \rightarrow Y$ given limited data, where X and Y denote the input feature and label, respectively. The function f is parameterized by a high-dimensional random vector z whose distribution is denoted as $P(z|C)$. This distribution is conditioned on the context data $C = \{(x_C, y_C)\}$ and estimated by an *encoder*. For instance, if we assume that the distribution of z follows a Gaussian distribution, the encoder would predict the mean and variance of this Gaussian distribution.

NP can readily adapt to new prediction tasks by sampling a z from the distribution. The prediction likelihood over *target data* $\mathcal{D} = \{(x_\mathcal{D}, y_\mathcal{D})\}$ is modeled as

$$P\left(y_\mathcal{D}|x_\mathcal{D}, C\right) = \int_z P\left(y_\mathcal{D}|x_\mathcal{D}, z\right) P(z|C) dz, \tag{1}$$

where $P\left(y_\mathcal{D}|x_\mathcal{D}, z\right)$ is modeled by a *decoder* network. Due to the intractable actual distribution of z, the optimization of the NP is achieved by amortized variational inference [24]. The objective function in Eq. (1) can be optimized by maximizing the evidence lower bound (ELBO), which is formulated as

$$\begin{aligned} \log P\left(y_\mathcal{D}|x_\mathcal{D}, C\right) \geq\ & \mathbb{E}_{Q_\theta(z|C,\mathcal{D})}\left[\log P_\phi\left(y_\mathcal{D}|x_\mathcal{D}, z\right)\right] \\ & - KL\left(Q_\theta(z|C, \mathcal{D}) \| P_\theta(z|C)\right), \end{aligned} \tag{2}$$

Table 1. Glossary of commonly used notations.

Notation	Definition
$\mathcal{G}$	knowledge graph
$\mathcal{E}, \mathcal{R}$	the sets of entities and relations
$\mathcal{T}$	a collection of triples
h, r, t	head entity, relation, and tail entity
C_{r_q}	context data (*a.k.a* support set) given a new relation r_q
$C^-_{r_q}$	negative triples for context data
$\mathcal{D}_{r_q}$	target data (*a.k.a* query set) given a new relation r_q
ρ, z	hypotheses extracted from context data
θ, ϕ, ψ	parameters of encoder, decoder, and extractor, respectively
$f_\rho(\cdot)$	function to check whether the hypothesis exists
$\mathcal{G}_{(h,t)}$	enclosing subgraph associated with h and r
$\mathcal{N}_k(h)$	k-hop neighbor triples of h
$\mathbb{1}(\cdot)$	Indicator function specifying whether two nodes are the same
$\|$	concatenation function
W, b, σ	learnable transformation matrix, bias, and nonlinear activation function
$\mathbf{h}_{\mathcal{G}_{(h,t)}}, \mathbf{h}_{\mathcal{G}_S}$	representation of subgraph
y_i	an indicator vector to denote whether the triple is positive or not
$\mathcal{N}(\mu(\mathbf{z}), \sigma(\mathbf{z}))$	the Gaussian distribution parameterized by $\mathbf{z}$
$\mathcal{G}_S$	subgraph identified by the hypothesis
M_z	soft edge mask
$\odot$	element-wise product
$g_\psi(\cdot)$	extractor to extract the subgraph $\mathcal{G}_S$ grounded by hypothesis z
$f_\phi(\cdot)$	measurement for the plausibility of the triple

where θ and ϕ represent the parameters of *encoder* and *decoder*, respectively, and $Q_\theta(z|C, \mathcal{D})$ denotes the estimation of the actual posterior distribution.

3.2 Graph Stochastic Attention

In graph learning tasks (*e.g.,* graph classification), the goal is to predict the label Y of a given graph $\mathcal{G}$ using a graph prediction function $f : \mathcal{G} \rightarrow Y$. Inspired by the principle of information bottleneck [54], graph stochastic

attention (GSAT) [40] aims to identify a subgraph $\mathcal{G}_S$ that mostly indicates the label Y by imposing an information constraint. This can be formulated as

$$\min_{\psi} -I\left(\mathcal{G}_S; Y\right) + \beta \cdot I\left(\mathcal{G}_S; \mathcal{G}\right), \text{ s.t. } \mathcal{G}_S \sim g_\psi(\mathcal{G}), \tag{3}$$

where $g_\psi(\mathcal{G})$ denotes the extractor with parameter ψ to extract possible subgraphs from $\mathcal{G}$, $I(\cdot,\cdot)$ indicates the mutual information (MI) between two random variables, and β controls the trade-off between the maximization of $I\left(\mathcal{G}_S; Y\right)$ and the minimization of $I\left(\mathcal{G}_S; \mathcal{G}\right)$. By optimizing Eq. (3), we encourage the $\mathcal{G}_S$ to inherit the relevant information in Y, while minimizing the irrelevant information from the input $\mathcal{G}$. This allows $\mathcal{G}_S$ to effectively reveal patterns for predictions and demonstrate explainability.

Following previous works [1], we can derive a tractable variational upper bound of Eq. (3), which is formulated as

$$\begin{aligned} \min_{\psi,\phi} -\mathbb{E}_{\mathcal{G}_S,Y}\left[\log P_\phi\left(Y|\mathcal{G}_S\right)\right] + \beta \cdot \mathbb{E}_{\mathcal{G}}\left[\mathrm{KL}(P_\psi(\mathcal{G}_S|\mathcal{G}) \| Q(\mathcal{G}_S))\right], \\ \text{s.t. } \mathcal{G}_S \sim P_\psi(\mathcal{G}_S|\mathcal{G}), \end{aligned} \tag{4}$$

where $P_\psi\left(\mathcal{G}_S|\mathcal{G}\right)$ denotes an *extractor* with parameter ψ that extracts the subgraph $\mathcal{G}_S$ from $\mathcal{G}$, and $P_\phi\left(Y|\mathcal{G}_S\right)$ denotes a *predictor* with parameter ϕ that predicts the label Y based on the subgraph $\mathcal{G}_S$. The $Q\left(\mathcal{G}_S\right)$ is the variational approximation for the marginal distribution of $\mathcal{G}_S$.

3.3 Problem Definition

A knowledge graph (KG) can be denoted as $\mathcal{G} = \{\mathcal{E}, \mathcal{R}, \mathcal{T}\}$, where $\mathcal{E}$ and $\mathcal{R}$ respectively represent the set of entities and relations. The $\mathcal{T} = \{(h, r, t) \subseteq \mathcal{E} \times \mathcal{R} \times \mathcal{E}\}$ represents a collection of triples, where h, r, and t denote the head entity, relation, and tail entity, respectively.

Conventional few-shot KG completion (FKGC) aims to predict missing facts for a new relation in the *query set* with a few-shot triple given in the *support set*. Previously, entities in the support set and query set are assumed to exist in the KGs. However, in real-world scenarios, new relations and entities may emerge simultaneously that are not included in the original background KGs. This paper focuses on this more challenging task, denoted as inductive few-shot KG completion (I-FKGC).

Inductive Few-shot Knowledge Graph Completion (I-FKGC). Given a new relation $r_q \notin \mathcal{R}$, and its associated K-shot support set $\left\{\left(h_i, r_q, t_i\right)\right\}_{i=1}^{K}$, we aim to predict the other entity t_q for each query in the query set $\left\{\left(h_q, r_q, t_q\right)\right\}$, where all the entities in the query set do not exist in the background knowledge graph, *i.e.,* $h_q, t_q \notin \mathcal{E}$.

In our paper, we propose a neural process-based framework for the I-FKGC task. Formally, given a new relation r_q, we treat its support set as the context data $\mathcal{C}_{r_q} = \left\{\left(h_i, r_q, t_i\right)\right\}_{i=1}^{K}$ and the query set as the target data $\mathcal{D}_{r_q} = \left\{\left(h_q, r_q, ?\right)\right\}$.

4 APPROACH

In this section, we first discuss the motivation from the inductive reasoning perspective, and then we introduce the proposed method.

4.1 I-FKGC as an Inductive Reasoning Problem

Inductive reasoning [18] is the process of formulating a general hypothesis based on past observations and applying it to predict future events. Inspired by this, we cast the inductive few-shot link prediction as an inductive reasoning problem. Specifically, we try to find a hypothesis (*i.e.,* shared subgraph of all triples) in the support set and then test whether the hypothesis can be applied to the query set for predictions.

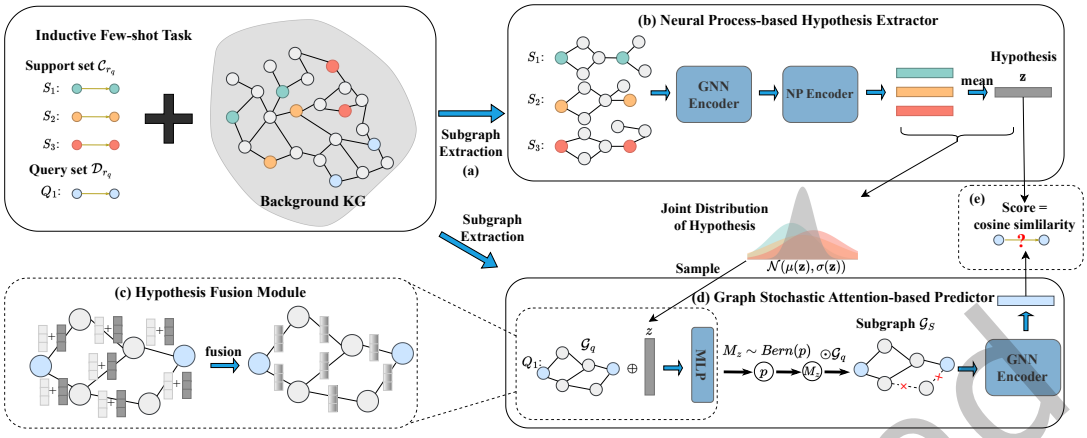

Fig. 2. The overall framework of our proposed model GS-NP, which consists of two major modules, namely, a neural process hypothesis extractor (including (b)) and a graph stochastic attention-based predictor (including (c), (d), and (e)). (a) We first extract enclosing subgraphs for all triples from the background knowledge graph. (b) We adopt the GNN encoder and NP encoder to model the joint distribution of the hypothesis. (c) After sampling a hypothesis (*i.e.*, z) from the distribution, we inject the hypothesis into the graph structure by the hypothesis fusion module. (d) We apply the graph stochastic attention to identify a subgraph and feed it into the GNN encoder to get representation. (e) We compute the cosine similarity between the subgraph representation and the hypothesis to test whether the query aligns with the shared hypothesis extracted from the support set.

Example. As the example shown in Figure 1(b), given a new relation r_q, we want to find a hypothesis that could explain how h establishes a relation r_q with the t for prediction. By examining the subgraph structure surrounding the triples, we can observe a shared connection pattern between h_k and t_k in the support set from background KGs $\mathcal{G}$. In this case, we can extract a hypothesis ρ from the support set as:

$$\rho := (h_q, \texttt{requires}, e_q) \wedge (e_q, \texttt{works in}, t_q) \rightarrow (h_q, r_q, t_q),$$

which means that if the hypothesis on the left-hand side is true, we can deduce the target relation between entities as (h_q, r_q, t_q). Thereby, the prediction function based on the hypothesis for I-FKGC can be formulated as $f_\rho : \rho(h_q, r_q, t_q) \rightarrow y, y \in \{0, 1\}$.

Based on the above motivation, we design the overall framework of GS-NP, as illustrated in Figure 2, for the I-FKGC task, which consists of two major modules, namely:

- **A neural process-based hypothesis extractor** to generate the hypothesis shared in the support set. To this end, we employ GNN and Neural Process (NP) encoders to capture the structure dependence of entities and model the joint distribution of the potential hypothesis, based on which we obtain a hypothesis to capture the shared pattern in the support set.
- **A graph stochastic attention-based predictor** to test whether the hypothesis can be applied to the query set for predictions. By injecting the extracted hypothesis into the graph structure, we develop a graph stochastic attention module to identify a subgraph from the query triple, whose representation will be compared with the extracted hypothesis, to determine the plausibility of triples.

We provide details on these two components below.

4.2 Neural Process-based Hypothesis Extractor

Neural process-based hypothesis extractor attempts to extract the inductive hypothesis from the support set for predictions. Existing methods extract the hypothesis by finding the most commonly shared subgraph structure among the support set [17]. However, due to the limited number of support triples, the extracted hypothesis may not generalize effectively to the query set. Drawing inspiration from the neural process, we propose to model the joint distribution of hypothesis. This allows us to sample a hypothesis from the distribution that generalizes well to the query set instead of overfitting to the sparse data.

4.2.1 Graph Neural Network Encoder. The graph structure inherently encodes the patterns for inductive reasoning [53]. Therefore, we extract the enclosing subgraph $\mathcal{G}_{(h,t)} \subset \mathcal{G}$ for each triple (h, r_q, t) in the support set, which can be formulated as

$$\mathcal{G}_{(h,t)} = \mathcal{N}_k(h) \cap \mathcal{N}_k(t), \tag{5}$$

where $\mathcal{N}_k(h)$, $\mathcal{N}_k(t)$ denotes the k-hop neighbor triples of h, t, respectively. The value of k is set to 1 or 2 according to the density of KGs [17]. The enclosing subgraph captures the relevant structural information about the entities h and t, which can be used to deduce the potential relation.

To extract the shared features of all the subgraphs in the support set, we adopt a GNN encoder called PathCon [59] to obtain the representation of each subgraph $\mathbf{h}_{\mathcal{G}_{(h,t)}}$. The PathCon focuses solely on the edge feature e_r through a relation message passing scheme, which enhances its ability to capture structural patterns for hypothesis extraction. This is formulated as

$$a_v^l = \sum_{(v,r,u)\in\mathcal{N}_1(v)} e_r^l, \tag{6}$$

$$e_v^l = a_v^l \| \mathbb{1}(v = h) \| \mathbb{1}(v = t), \tag{7}$$

$$e_r^{l+1} = \sigma\left(\left(e_v^l \| e_u^l \| e_r^l\right) \cdot W^l + b^l\right), (v, r, u) \in \mathcal{G}_{(h,t)}, \tag{8}$$

where "$\|$" is the concatenation function, a_v^l is the representation of entity v at layer l, $(v, r, u) \in \mathcal{N}_1(v)$ denotes all the triples directly connecting to v, e_r^l is the hidden representation of edge r. The e_r^1 can be initialized randomly or from pretrained relation embedding and e_v^l denotes the hidden representation of entity v. The operation $\mathbb{1}(\cdot)$ indicates whether two nodes are the same and returns 1 if same and 0 for different, which injects the information of target entities h, t into the subgraph representation. W^l, b^l, and σ denote the learnable transformation matrix, bias, and nonlinear activation function.

By stacking L layers, we can obtain the subgraph representation $\mathbf{h}_{\mathcal{G}_{(h,t)}}$ by summarizing all the edge representations with max pooling and concatenating them with target entities representations as

$$\mathbf{e} = \text{MaxPooling}(e_r^L | r \in \mathcal{G}_{(h,t)}), \tag{9}$$

$$\mathbf{h}_{\mathcal{G}_{(h,t)}} = \mathbf{e} \| a_h^L \| a_t^L. \tag{10}$$

4.2.2 Neural Process Encoder. The $\mathbf{h}_{\mathcal{G}_{(h,t)}}$ encodes the inductive features that can be used to deduce the relation. Therefore, the neural process encoder tries to capture connections between the hidden representation and the target relation from the support set to model the distribution of the hypothesis $P(\mathbf{z}|C_{r_q})$.

For each triple (h_i, r_q, t_i) in the support set, we first get its subgraph representation $\mathbf{h}_{\mathcal{G}_{(h_i,t_i)}}$ by GNN encoder. To reduce the estimation bias of the distribution, we generate a few negative triples in the support set as $C_{r_q}^-$ by corrupting the head or tail entities [58]. Then, we obtain their subgraph representations with the same GNN encoder.

To capture the connections for representing the hypothesis, we concatenate the subgraph representation with an indicator vector y_i to generate c_i as follows:

$$c_i = \text{MLP}\left(\mathbf{h}_{\mathcal{G}_{(h_i,t_i)}} \| y_i\right), y_i = \begin{cases} 1, \left(h_i, r_q, t_i\right) \in C_{r_q} \\ 0, \left(h_i, r_q, t_i\right) \in C_{r_q}^- \end{cases}, \tag{11}$$

where y_i denotes whether the triple is positive or not.

Then, we summarize all the latent representations $c_i \in C_{r_q} \cup C_{r_q}^-$ into a global representation $\mathbf{z}$ by a *permutation-invariant* aggregator function [12, 55], to model the joint distribution over hypothesis. In our paper, we select a simple average function, which is formulated as

$$\mathbf{z} = \frac{1}{\left|C_{r_q} \cup C_{r_q}^-\right|} \sum_{c_i \in C_{r_q} \cup C_{r_q}^-} c_i. \tag{12}$$

The distribution $P(\mathbf{z}|C_{r_q})$ is empirically assumed to follow the Gaussian distribution $\mathcal{N}(\mu(\mathbf{z}), \sigma(\mathbf{z}))$, which is parameterized by $\mathbf{z}$. The mean $\mu(\mathbf{z})$ and variance $\sigma(\mathbf{z})$ can be estimated by using two neural networks as follows

$$\chi = \text{ReLU}(\text{MLP}(\mathbf{z})), \tag{13}$$

$$\mu(\mathbf{z}) = \text{MLP}\left(\chi\right), \tag{14}$$

$$\sigma(\mathbf{z}) = 0.1 + 0.9 \cdot \text{Sigmoid}\left(\text{MLP}\left(\chi\right)\right). \tag{15}$$

Finally, we can sample a z from the distribution as the instance of the hypothesis, which is formulated as

$$\text{Sample } z \sim \mathcal{N}(\mu(\mathbf{z}), \sigma(\mathbf{z})), \tag{16}$$

where z is considered as the hidden representation of the hypothesis, which will be used for predictions.

4.3 Graph Stochastic Attention-based Predictor

Graph stochastic attention-based predictor is designed to test the extracted hypothesis on the query for predictions. However, as the extracted hypothesis z is a latent variable, it cannot be directly applied to the graph structure. To solve this problem, we employ the graph stochastic attention mechanism [9] to transform the hypothesis z into a soft edge mask M_z. This mask allows us to extract a subgraph $\mathcal{G}_S$ identified by the hypothesis, which can then be used for prediction and offering explanations.

For each triple in the query set $\mathcal{D}_{r_q}$, we first extract its enclosing subgraph $\mathcal{G}_{(h,t)}$ by Eq. (5). Following the paradigm of graph stochastic attention, we attempt to learn an *extractor* g_ψ to model the distribution $P_\psi(\mathcal{G}_S|\mathcal{G}_{(h,t)}, z)$ and extract a subgraph $\mathcal{G}_S$ grounded by hypothesis z. Specifically, we assume the distribution $P_\psi(\mathcal{G}_S|\mathcal{G}_{(h,t)}, z)$ following the Bernoulli distribution [40], which is formulated by the joint probability of all edges in the subgraph $\mathcal{G}_S$ as follows

$$P_\psi(\mathcal{G}_S|\mathcal{G}_{(h,t)}, z) = \prod_{(h,r,t) \in \mathcal{G}_{(h,t)}} P_\psi(r|e_r, z), \tag{17}$$

$$P_\psi(r|e_r, z) = \text{Sigmoid}(\text{MLP}_\psi(e_r + z)), \tag{18}$$

where $P_\psi(r|e_r, z) \in [0, 1]$ denotes the existence possibility of each edge in $\mathcal{G}_S$, which is estimated by the *hypothesis fusion module*, and "+" denotes the element-wise addition. MLP_ψ and Sigmoid denote the multi-layer perception with parameter ψ and an activation function, respectively. Subgraph $\mathcal{G}_{(h,t)}$ refers to a subgraph induced by the entities h and t for a given triple in the dataset. Specifically, this subgraph is constructed by retaining only the common neighbors of entities h and t, while $\mathcal{G}_S$ denotes the subgraph identified by GSAT from $\mathcal{G}_{(h,t)}$.

Algorithm 1: The training process of GS-NP

Input: Knowledge graph $\mathcal{G}$; Training relations $\mathcal{R}_{train}$
Output: encoder parameters: θ, decoder parameters: ϕ, extractor parameters: ψ

Initialization: θ, ϕ, and ψ randomly;
while *not converge* **do**
  Sample a relation $\mathcal{T}_{r_q} = \{C_{r_q}, \mathcal{D}_{r_q}\}$ from $\mathcal{R}_{train}$;
  Extract the enclosing subgraph $\mathcal{G}_{(h,t)}$ for each triple $(h, r_q, t) \in \mathcal{T}_{r_q}$ (Eq. (5));
  Generate the subgraph representation $\mathbf{h}_{\mathcal{G}_{(h,t)}}$ (Eqs. (6)-(10));
  Generate the prior distribution $P(z|C_{r_q})$ by C_{r_q} (Eqs. (11)-(15));
  Generate the variational posterior distribution $Q_\theta\left(z|C_{r_q}, \mathcal{D}_{r_q}\right)$ by $C_{r_q}, \mathcal{D}_{r_q}$ (Eqs. (11)-(15));
  Sample a z from the posterior distribution $Q_\theta\left(z|C_{r_q}, \mathcal{D}_{r_q}\right)$ (Eq. (16));
  Extract a subgraph $\mathcal{G}_S$ given the z, $\mathcal{G}_{(h,t)}$ (Eqs. (17)-(20));
  Calculate the similarity between $\mathcal{G}_S$ and z (Eq. (21));
  Optimize θ, ϕ, and ψ using ELBO loss (Eq. (26));
end

By sampling from $P_\psi(\mathcal{G}_S|\mathcal{G}_{(h,t)}, z)$, we can obtain an edge mask M_z to extract the subgraph as

$$\text{Sample } M_z \sim \text{Bern}(P_\psi(\mathcal{G}_S|\mathcal{G}_{(h,t)}, z)), \tag{19}$$

$$\mathcal{G}_S = \mathcal{G}_{(h,t)} \odot M_z, \tag{20}$$

where "$\odot$" denotes the element-wise product between edges and mask. To ensure the computable gradient of $P_\psi(r|e_r, z)$, we apply the gumbel-softmax reparameterization [19] to sample mask M_z.

The extracted subgraph $\mathcal{G}_S$ is then fed into the *predictor* f_ϕ to test whether the query set aligns with the hypothesis. We first adopt the same GNN encoder in the hypothesis extractor to obtain the subgraph representation $\mathbf{h}_{\mathcal{G}_S}$. Then we compute the similarity (*e.g.,* cosine similarity) between the subgraph representation $\mathbf{h}_{\mathcal{G}_S}$ and the hypothesis z to measure the plausibility of the triple, which can be formulated as

$$f_\phi(\mathcal{G}_S, z) = \text{Cosine}(\mathbf{h}_{\mathcal{G}_S}, z). \tag{21}$$

4.4 Optimization

GS-NP draws inspiration from both neural processes and graph stochastic attention. In this section, we present the optimization process of GS-NP, which seamlessly combines the optimization of neural processes and graph stochastic attention into a unified objective function. The training process of GS-NP is illustrated in Algorithm 1.

Given a relation r_q and corresponding support set and query set, GS-NP aims to extract a hypothesis z from the support set C_{r_q} and minimize the prediction loss on the query set $\mathcal{D}_{r_q}$. This can be optimized by maximizing

the evidence lower bound (ELBO), which can be derived as

$$\begin{aligned}
&\log P\left(t_q|h_q, r_q, C_{r_q}, \mathcal{G}_q\right) \geq \int_Z Q(Z) \log \frac{P(t_q, Z|h_q, r_q, C_{r_q}, \mathcal{G}_q)}{Q(Z)} \\
&= \mathbb{E}_{Q(Z)}\left[\log P(t_q|h_q, r_q, Z) + \log \frac{P(\mathcal{G}_S, z|h_q, r_q, C_{r_q}, \mathcal{G}_q)}{Q(Z)}\right] \\
&= \underbrace{\mathbb{E}_{Q(Z)}\left[\log P\left(t_q|h_q, r_q, Z\right)\right]}_{(1)} \underbrace{- KL(Q(z)\|P(z|C_{r_q}))}_{(2)} \\
&\underbrace{- KL(Q(\mathcal{G}_S)\|P(\mathcal{G}_S|\mathcal{G}_q, z))}_{(3)},
\end{aligned} \tag{22}$$

where $Z = (\mathcal{G}_S, z)$ denotes all the latent variables, and $\mathcal{G}_q$ denotes the original enclosing subgraph of the query $\mathcal{G}_{(h_q,t_q)}$ for simplicity. The hypothesis z depends on the support set C_{r_q} and extracted subgraph $\mathcal{G}_S$ depends on the query set $\mathcal{D}_{r_q}$, support set C_{r_q}, and hypothesis z. The $Q(z)$ is the true posterior distribution of z, which is approximated by $Q_\theta(z|C_{r_q}, \mathcal{D}_{r_q})$ during training. The $Q(\mathcal{G}_S)$ denotes the variational approximation to the true distribution over the subgraph. The true posterior distribution z and $\mathcal{G}_S$ (*a.k.a.* $Q(z)$ and $Q(\mathcal{G}_S)$) are independent from each other. The derivation of the ELBO loss in Eq. (22) can be found in the Appendix A.

From Eq. (22), we can see that the combination of part (1) and part (2) is identical to the objective of the neural process, while the combination of part (1) and part (3) is the objective of graph stochastic attention. Therefore, the Eq. (22) seamlessly combined the objectives of these two modules which are jointly optimized during training.

According to the GSAT [40], the third KL-divergence term in Eq. (22) can be further computed as

$$\begin{aligned}
&\mathrm{KL}\left(Q\left(\mathcal{G}_S\right)\|P\left(\mathcal{G}_S|\mathcal{G}_q, z\right)\right) = \\
&\sum_{(h,r,t)\in\mathcal{G}_q} p_{e_r} \log \frac{p_{e_r}}{\tau} + \left(1 - p_{e_r}\right) \log \frac{1 - p_{e_r}}{1 - \tau} + c(n, \tau),
\end{aligned} \tag{23}$$

where p_{e_r} is the probability predicted by hypothesis fusion module, n denotes the number of edges in $\mathcal{G}_q$, $\tau \in [0, 1]$ is a hyper-parameter, and $c(n, \tau)$ is a constant value.

The final objective function is presented as follows:

$$\begin{aligned}
&\mathcal{L}(\phi, \theta) = \mathbb{E}_{Q(\mathcal{G}_S,z)}\left[\log P_\phi\left(t_q|h_q, r_q, \mathcal{G}_S, z\right)\right] \\
&- \mathrm{KL}(Q_\theta(z|C_{r_q}, \mathcal{D}_{r_q})\|P_\theta(z|C_{r_q})) - \mathrm{KL}(Q(\mathcal{G}_S)\|P(\mathcal{G}_S|\mathcal{G}, z)),
\end{aligned} \tag{24}$$

where θ and ϕ denote the parameters of *encoder* and *predictor* (*a.k.a., decoder*).

To support gradient propagation, we introduce the *reparamterization trick* for sampling z, and then we estimate the expectation $\mathbb{E}_{Q(\mathcal{G}_S,z)}\left[\log P_\phi\left(t_q|h_q, r_q, \mathcal{G}_S, z\right)\right]$ via the Monte-Carlo sampling as

$$\begin{aligned}
&\mathbb{E}_{Q(\mathcal{G}_S,z)}\left[\log P_\phi\left(t_q|h_q, r_q, \mathcal{G}_S, z\right)\right] \simeq \frac{1}{T}\sum_{t=1}^{T} \log P_\phi\left(t_q|h_q, r_q, \mathcal{G}_S, z^{(t)}\right), \\
&z^{(t)} = \mu(\mathbf{z}) + \sigma(\mathbf{z})\epsilon^{(t)}, \text{ with } \epsilon^{(t)} \sim \mathcal{N}(0, 1).
\end{aligned} \tag{25}$$

The first term in Eq. (24) represents the prediction likelihood of both the decoder in NPs and the predictor in graph stochastic attention. This can be calculated using the commonly used margin ranking loss, which is

$$\log P_\phi(t_q|h_q, r_q, \mathcal{G}_S, z) = -\sum_{q,q^-} \max(0, \gamma + s(q^-) - s(q)), \tag{26}$$

where γ denotes a margin hyper-parameter, and q, q^- represent true triples and negative triples, respectively. We attempt to rank the scores of true triples higher than all other negative triples by maximizing the likelihood.

4.5 Complexity Analysis

In this section, we will concisely outline the complexity analysis of our method. Given a relation r_q with its K-shot support set C_{r_q} and m factual triples in the query set $\mathcal{D}_{r_q}$, the complexity of GS-NP is $O((n+1)K+m)$, where n denotes the negative sampling size for each triple in C_{r_q}. Our method only needs to encode each triple in $C_{r_q} \cup C_{r_q}^-$ and predict triples in the query set.

5 EXPERIMENTS

In this section, we will qualitatively and quantitatively demonstrate the superiority of our method.

5.1 Datasets and Evaluation

Table 2. Statistics of datasets. Ind-BG refers to the background KG used in training time and Ind-Test refers to the knowledge graph constructed by all entities and triples involved in the test tasks.

KG	NELL				ConceptNet				WIKI			
	#rels	#entities	#edges	#tasks	#rels	#entities	#edges	#tasks	#rels	#entities	#edges	#tasks
Ind-BG	291	44,005	82,318	-	14	619,163	1,191,782	-	822	2,583,905	3,221,617	-
Ind-Test	291	24,539	98,791	11	14	171,540	1,350,214	2	822	2,179,254	2,637,623	33

We conduct evaluations on three widely used FKGC benchmarks, namely, NELL [41], ConceptNet [52], and WIKI [56]. Following the setting of CSR [17], we construct the datasets in an *inductive manner*. We remove all entities in the test tasks, together with their one-hop neighbors, from the original KGs to construct an inductive background KG (*Ind-BG*) during training. All entities and triples involved in the test tasks become the *Ind-Test*. In the test time, we combine *Ind-Test* with *Ind-BG* to construct a test time background KG.

Specifically, for the inductive setting, we mostly use the meta-eval and meta-test splits of NELL-One for the eval and test few-shot tasks on NELL and select the fewest 1/2 appearing relations as eval/test few-shot tasks for the ConceptNet, following the previous paper [39, 62]. For each test task, we also subsample the number of query triplets to 10%. Following the CSR, for the WIKI dataset, we first remove all entities and their one-hop neighbors involved in test tasks from the original knowledge graph and then subsample the number of query triples to 2% to make sure that the remaining training time background KG does not become too small. The statistics of datasets are summarized in Table 2.

In the test time, we sample 50 negative tail candidates for each query triple and rank them together with the true tail entity. We adopt two widely used metrics, namely mean reciprocal rank (MRR) and the Top-N hit ratio (Hit@N). Hit@N measures the percentage of times that the positive tail is ranked higher than N among the negative tail candidates. The N is set to 1, 5, and 10 to fairly compare with existing methods [17].

5.2 Baseline Models

We compare the proposed method with two groups of approaches: **FKGC Methods**, including GMatching [62], MetaR [5], FSRL [65], FAAN [50], GANA [42], NP-FKGC [36], RelAdapter [47], and ReCDAP-FKGC [22]; and **Inductive KGC Methods**, which include PathCon [59], SNRI [64], GraIL [53], and CSR [17]. CSR is the current state-of-the-art (SOTA) method designed for the I-FKGC task. The details of baseline models are shown as follows.

FKGC methods. This group of methods is all under the meta-learning framework, which can predict potential links given few-shot associated triples.

- GMatching[1] [62] utilizes the entity embeddings and entity local graph structures to represent entity pairs and learn a metric to measure the similarity, which is the first work for FKGC.
- FAAN[2] [50] adopts an adaptive neighbor encoder to encode entities and a Transformer to encoder entity pairs.
- MetaR[3] [5] attempts to learn relation-specific meta information from support triples and applies it to few-shot relational predictions.
- GANA[4] [42] introduces a gated and attentive neighbor aggregator to represent entity representations.
- NP-FKGC[5] [36] integrates a stochastic ManifoldE encoder to incorporate the neural process and handle complex few-shot relations.
- RelAdapter[6] [47] is context-aware adapter that enhances meta-learning for few-shot relation learning in knowledge graphs by enabling relation-specific adaptation with lightweight modules and contextual information.
- ReCDAP[7] [22] integrates positive information from the KG and non-existent negative information into the diffusion process, which enables the model to explicitly separate and accurately estimate positive and negative embedding distributions.

Inductive KGC methods. This group of methods can predict missing factual triples inductively.

- PathCon[8] [59] combines two types of subgraph structures (*i.e.,* the contextual relations and the relational paths between entities) to predict link inductively.
- SNRI[9] [64] leverages the complete neighbor relations of entities using extracted neighboring relational features and paths to handle sparse subgraphs.
- GraIL[10] [53] is a GNN-based method to extract enclosing subgraph between two unseen entities for inductive link prediction.
- CSR[11] [17] uses a connection subgraphs to represent entity pairs and test whether each triple in query set aligns with shared connection subgraph extracted from support set.

5.3 Implementation Details

We implement our model with PyTorch and PyG packages and conduct our experiments on a single RTX 3090 GPU. For the GNN encoder, we adopt a 3-layer PathCon with the hidden dimension set to 128. The dimension of subgraph representation and hypothesis z is set to 100 for NELL as well as ConceptNet, and 50 for WIKI. The τ in stochastic attention is set to 0.7. We use Adam [23] as the optimizer. The negative sampling size is set to 1, the learning rate is set to 10^{-5}, and the margin γ is set to 1. The best model used for testing is selected by the MRR metric on the validation dataset. For baseline methods, we implement these methods from the repositories publicized by their authors. For the I-FKCG problem, we initialize entity embeddings randomly for our methods as they are unknown during testing. All methods use 100-dimensional relation and entities embedding for NELL and ConceptNet datasets and 50 dimensions for the WIKI dataset when applicable.

[1]https://github.com/xwhan/One-shot-Relational-Learning
[2]https://github.com/JiaweiSheng/FAAN
[3]https://github.com/AnselCmy/MetaR
[4]https://github.com/ngl567/GANA-FewShotKGC
[5]https://github.com/RManLuo/NP-FKGC
[6]https://github.com/smufang/RelAdapter
[7]https://github.com/hou27/ReCDAP-FKGC
[8]https://github.com/hwwang55/PathCon
[9]https://github.com/Tebmer/SNRI
[10]https://github.com/kkteru/grail
[11]https://github.com/snap-stanford/csr

Table 3. The results of 3-shot I-FKGC on NELL, ConceptNet, and WIKI datasets.

Method	NELL				ConceptNet				WIKI			
	MRR	**Hit@1**	**Hit@5**	**Hit@10**	**MRR**	**Hit@1**	**Hit@5**	**Hit@10**	**MRR**	**Hit@1**	**Hit@5**	**Hit@10**
PathCon	0.003	0.000	0.000	0.000	0.063	0.000	0.000	0.000	0.003	0.000	0.000	0.000
GraIL	0.077	0.049	0.070	0.070	0.046	0.009	0.016	0.016	0.026	0.000	0.000	0.000
SNRI	0.195	0.107	0.247	0.291	0.193	0.138	0.190	0.231	0.003	0.000	0.000	0.000
GMatching	0.288	0.152	0.427	0.601	0.283	0.151	0.438	0.548	0.151	0.059	0.204	0.340
MetaR	0.251	0.135	0.388	0.494	0.269	0.151	0.411	0.493	0.110	0.038	0.137	0.244
FSRL	0.204	0.124	0.208	0.236	0.285	0.164	0.370	0.493	0.235	0.163	0.271	0.326
FAAN	0.485	0.326	0.691	0.803	0.210	0.096	0.274	0.438	0.123	0.076	0.122	0.145
GANA	0.297	0.163	0.461	0.573	0.239	0.137	0.329	0.397	0.118	0.046	0.115	0.221
NP-FKGC	0.354	0.152	0.612	0.787	0.169	0.014	0.274	0.630	0.613	0.550	0.673	0.712
RelAdapter	0.213	0.101	0.309	0.472	0.269	0.164	0.342	0.466	0.103	0.031	0.0093	0.256
ReCDAP	0.290	0.180	0.399	0.517	0.478	0.315	0.699	0.945	0.618	0.465	**0.837**	**0.891**
CSR	0.511	0.348	0.725	0.837	0.611	0.496	**0.729**	**0.786**	0.666	0.589	0.744	0.798
GS-NP	**0.627**	**0.522**	**0.742**	**0.865**	**0.634**	**0.548**	0.712	0.740	**0.694**	**0.597**	0.822	0.860

Existing FKGC methods can generalize to new relations but not to unseen entities since the entity representations are constructed from a set of embeddings pretrained on background KGs. Metric learning methods such as GMatching, FSRL, and FAAN rely on encoding embeddings of entity pairs to measure the similarity between the support set and query set. Meta-optimization-based methods (*e.g.,* MetaR, GANA) also require entity embeddings in the score function for prediction. However, under the I-FKGC setting where entities in the test phase are unseen, these methods experience significant performance degradation due to the lack of entity embeddings. Therefore, following previous studies [17], we violate the inductive setting and use pre-trained entity embeddings (*e.g.,* TransE [3]) for these FKGC methods to provide comparable results.

Inductive KGC methods are primarily designed for inductive settings and are not directly applicable to few-shot tasks. Consequently, adhering to their original configurations, we concatenate all meta-training tasks for the training phase and evaluate these methods across all query sets in the meta-testing phase for comparison.

5.4 Results and Analysis

As shown in Table 3, we report the results of 3-shot I-FKGC on the NELL, ConceptNet, and WIKI datasets. The best results are highlighted in bold and the second-best results are underlined. From the results, we can see that GS-NP outperforms all baseline methods and derives new SOTA performance on most metrics, which demonstrates the effectiveness of our model.

Traditional inductive KGC methods get the worst results when compared with all baseline methods. Despite being designed for inductive reasoning, they overlook the few-shot settings, causing them to struggle with generalizing to unseen relations with few-shot triples. Early FKGC methods (*e.g.,* GMatching, MetaR, FSRL) achieve better performance as they solve the problem in a meta-learning framework, which effectively utilizes the limited data to derive new knowledge. However, these methods often encounter out-of-distribution and overfitting issues, resulting in suboptimal results. To tackle this problem, NP-FKGC adopts the neural process to model the distribution of prediction functions and achieves improved performance. This demonstrates the effectiveness of the neural process in few-shot settings.

CSR casts the I-FKGC problem as an inductive reasoning problem, which attempts to extract the commonly shared graph structure as a hypothesis for predictions. As a result, CSR demonstrates good generalization to unseen entities and relations despite limited data, achieving the second-best performance among all baseline methods. Following the paradigm of inductive reasoning, GS-NP also extracts the hypothesis from the support

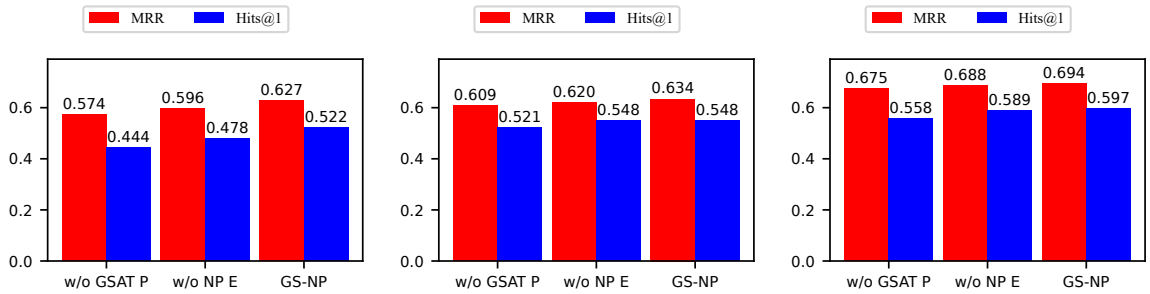

Fig. 3. Ablation study results on NELL, ConceptNet, and WIKI datasets.

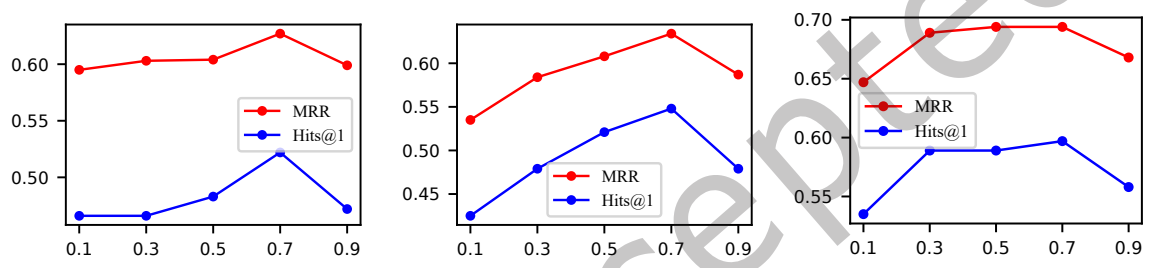

Fig. 4. Parameter analysis of r on NELL, ConceptNet, and WIKI datasets.

set by modeling the distribution of the hypothesis. By doing so, GS-NP combines both the merits of neural process and inductive reasoning. Moreover, GS-NP adopts a graph stochastic attention-based predictor to extract a label-relevant subgraph identified by the hypothesis for predictions. As a result, GS-NP outperforms CSR on most metrics, achieving the best overall performance among all baseline methods. For ConceptNet dataset, while our method achieves lower performance in terms of Hit@5 and Hit@10, it significantly outperforms previous methods on MRR and Hit@1 metrics, demonstrating that our approach is particularly effective in providing highly accurate top-ranked predictions.

5.5 Ablation Study

To evaluate the effectiveness of the neural process-based hypothesis extractor (**NP Extractor**) and graph stochastic attention-based predictor (**GSAT Predictor**), we perform an ablation study by removing each component of GS-NP. The experiments are conducted on the NELL, ConceptNet, and WIKI datasets with a 3-shot support set. From the results shown in Figure 3, where we abbreviate **NP Extractor** and **GSAT Predictor** as **NP E** and **GSAT P**, we can observe that all components can help improve the performance of our model. By removing the NP extractor, the extracted hypothesis is the average graph representation of the support set, which may not generalize well to the query set. Without the GSAT Predictor, the performance drops a lot as the extracted hypothesis is a latent variable. Applying it directly to graph structure impairs the performance and lacks explanations. To further ensure the reliability of the results, we conducted the ablation study three times and reported the mean and standard deviation in Appendix B.

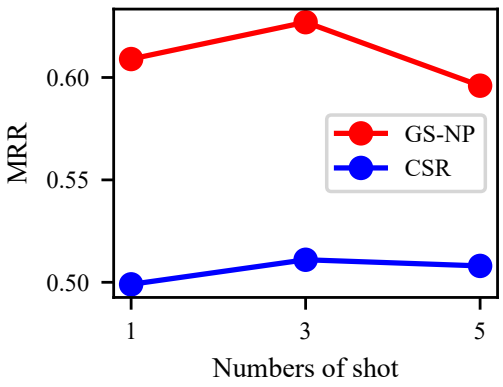

Fig. 5. MRR metric under different K-shot settings on NELL dataset.

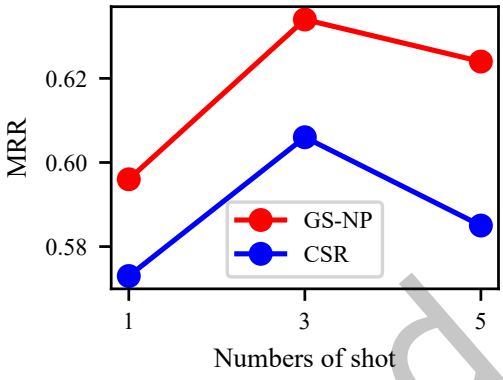

Fig. 6. MRR metric under different K-shot settings on ConceptNet dataset.

Table 4. Results of 3-shot transductive FKGC on NELL and ConceptNet datasets.

Method	NELL				ConceptNet			
	MRR	**Hit@1**	**Hit@5**	**Hit@10**	**MRR**	**Hit@1**	**Hit@5**	**Hit@10**
MetaR	0.471	0.322	0.647	0.763	0.318	0.226	0.390	0.496
FSRL	0.490	0.327	0.695	0.853	0.577	0.469	0.695	0.753
NP-FKGC	0.509	0.388	0.635	0.736	0.265	0.139	0.316	0.644
CSR	0.577	0.442	0.746	**0.858**	0.606	0.495	**0.735**	**0.777**
GS-NP	**0.617**	**0.495**	**0.771**	0.857	**0.608**	**0.512**	0.720	0.755

5.6 Parameter Analysis

We study the impact of the selection of τ in Eq. (23). From Figure 4, we can see that results decrease as τ decreases to 0. The best performance is achieved when τ is around 0.7. This is consistent with the theory observed by GSAT [40] that sets τ to a value close to 1 can extract a much denser subgraph and often provide a more robust interpretation.

To further study the ability of inductive reasoning, we conduct experiments under different K-shot scenarios compared to the existing SOTA method CSR. Figure 5 and Figure 6 illustrate the MRR metric under different K-shot scenarios on the NELL and ConceptNet datasets. As the value of K decreases to 1, the performance of CSR and GS-NP decline, and GS-NP is better than CSR. This suggests that by modeling the distribution of the hypothesis, our method can perform well even with just one triple. However, as K increases to 5, the performance of CSR and our method drops. One possible explanation for this is that when more triples in the support set are given, noise is introduced, which impairs the performance of both CSR and our method.

5.7 A Study on Transductive Setting

To further demonstrate the effectiveness of our method, we also test the performance of GS-NP under a transductive setting in the NELL and ConceptNet datasets, where the entities are seen during training. For baseline methods,

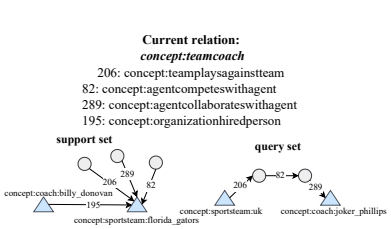

Fig. 7. Illustration of extracted subgraph for a specific relation *concept:teamcoach* in NELL dataset.

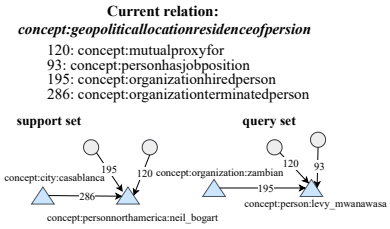

Fig. 8. Illustration of extracted subgraph for a specific relation *concept:geopoliticallocationresidenceofpersion* in NELL dataset.

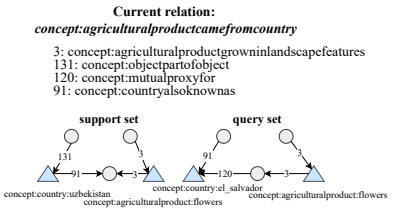

Fig. 9. Illustration of extracted subgraph for a specific relation *concept:agriculturalproductcamefrom country* in NELL dataset.

we directly use the results reported by CSR [17]. From the results shown in Table 4, we can observe that GS-NP can also be applied to transductive settings and achieve the best performance among all baseline methods. This is because GS-NP can extract meaningful hypotheses from subgraphs which can effectively be used to derive new knowledge.

5.8 Explanatory Subgraph for Predictions

The stochastic attention mechanism in GS-NP can generate an edge mask to extract a subgraph identified by the hypothesis. The extracted subgraph visualizes the hypothesis and explains the predictions. In Figure 7, we illustrate the extracted subgraph for a specific relation `concept:teamcoach` in the NELL dataset. From the visualization, it is evident that our method extracts a meaningful hypothesis from the support set, *e.g.*, `concept:teamplaysagainstteam`, `concept:agentcompeteswithagent`, `concept:agentcollaborateswithagent` $\Rightarrow$ `concept:teamcoach`. All three relations in the hypothesis can be found in the subgraph of the support set, indicating that our method effectively captures structure dependence from limited data. Based on the hypothesis, we can infer a potential relation `concept:teamcoach` between the head entity `concept:sportsteam:uk` and the tail entity `concept:sportsteam:uk` in the query set.

As shown in Figure 8, we illustrate the relation `concept:geopoliticallocationresidenceof persion` while as shown in Figure 9, we illustrate the relation `concept:agriculturalproductcame fromcountry`.

6 CONCLUSION

In this paper, we propose a novel graph stochastic neural process approach, called GS-NP for inductive few-shot knowledge graph completion (I-FKGC). Our method is motivated by the concept of inductive reasoning. To extract a general hypothesis, we employ a neural process-based hypothesis extractor to capture the structure dependence and model the joint distribution of the hypothesis, based on which we obtain a hypothesis to represent the shared pattern within the support set. We then use a graph stochastic attention-based predictor to inject the extracted hypothesis into the graph structure and identify a subgraph from the query for the predictions. In this way, we can not only test the applicability of the hypothesis but also provide explanatory predictions. Extensive experiments on three benchmark datasets demonstrate the effectiveness of our method.

In the future, we plan to unify large language models (LLMs) and knowledge graphs to improve the performance for inductive few-shot knowledge graph completion [44, 60]. Specifically, we will utilize the LLMs to enrich representations of KGs by encoding the textual descriptions of entities and relations, which facilitates the downstream tasks [6]. Meanwhile, we will also explore the potential of KGs to enhance LLMs' faithful and

interpretable reasoning on complex tasks [35, 37, 38, 45, 67]. Besides, we will try to explore the potential of GS-NP for temporal knowledge graph completion tasks [61].

ACKNOWLEDGMENTS

This research is supported by NSF of China (Nos: 62336003, 12371510).

APPENDIX

A DERIVATION OF ELBO LOSS

Given a relation r_q and its associated support set C_{r_q}, the objective of GS-NP is to infer the distributions $P(Z|C_{r_q}, \mathcal{G}_q)$ from the support set that minimizes the prediction loss on the query set $\mathcal{D}_{r_q}$. The unified training objective function for GS-NP can be formulated as

$$\begin{aligned} &P\left(t_q, Z|h_q, r_q, C_{r_q}, \mathcal{G}_q\right) \\ &= P(Z|C_{r_q}, \mathcal{G}_q) \prod_{\{(h_q, r_q, ?)\}} P(t_q|f_{r_q}(h_q, r_q, Z)), \end{aligned} \tag{A.1}$$

where $f_{r_q}(h_q, r_q, Z)$ denotes the *predictor*, $\{(h_q, r_q, ?)\}$ denotes the query to be predicted, and $Z = (\mathcal{G}_q, z)$ denotes all the latent variables.

Following the Eq. (A.1), the prediction likelihood on the query set $P(t_q|h_q, r_q, C_{r_q}, \mathcal{G}_q)$ can be written as

$$\begin{aligned} \log P(t_q|h_q, r_q, C_{r_q}, \mathcal{G}_q) &= \log \frac{P\left(t_q, Z|h_q, r_q, C_{r_q}, \mathcal{G}_q\right)}{P\left(Z|C_{r_q}, \mathcal{G}_q\right)} \\ &= \log P\left(t_q, Z|h_q, r_q, C_{r_q}, \mathcal{G}_q\right) - \log P\left(Z|C_{r_q}, \mathcal{G}_q\right). \end{aligned} \tag{A.2}$$

Assuming that $Q(Z)$ is the true distribution of Z, we can rewrite the Eq. (A.2) as

$$\begin{aligned} &\log P(t_q|h_q, r_q, C_{r_q}, \mathcal{G}_q) \\ &= \log \frac{P\left(t_q, Z|h_q, r_q, C_{r_q}, \mathcal{G}_q\right)}{Q(Z)} - \log \frac{P\left(Z|C_{r_q}, \mathcal{G}_q\right)}{Q(Z)}. \end{aligned} \tag{A.3}$$

Then, we integrate both sides with $Q(Z)$, and obtain

$$\begin{aligned} &\log P(t_q|h_q, r_q, C_{r_q}, \mathcal{G}_q) \\ &= \int_Z Q(Z) \log \frac{P\left(t_q, Z|h_q, r_q, C_{r_q}, \mathcal{G}_q\right)}{Q(Z)} - \int_Z Q(Z) \log \frac{P\left(Z|C_{r_q}, \mathcal{G}_q\right)}{Q(Z)} \\ &= \int_Z Q(Z) \log \frac{P\left(t_q, Z|h_q, r_q, C_{r_q}, \mathcal{G}_q\right)}{Q(Z)} + KL\left(Q(Z) \| P\left(Z|C_{r_q}, \mathcal{G}_q\right)\right). \end{aligned} \tag{A.4}$$

Since $KL(Q(Z)\|P\left(Z|C_{r_q},\mathcal{G}_q\right) \geq 0$, we can derive Eq. (A.4) as

$$
\begin{aligned}
&\log P(t_q|h_q,r_q,C_{r_q},\mathcal{G}_q) \geq \int_Z Q(Z)\log\frac{P\left(t_q,Z|h_q,r_q,C_{r_q},\mathcal{G}_q\right)}{Q(Z)}\\
&=\mathbb{E}_{Q(Z)}\log\frac{P\left(t_q,Z|h_q,r_q,C_{r_q},\mathcal{G}_q\right)}{Q(Z)}\\
&=\mathbb{E}_{Q(Z)}\left[\log P\left(t_q|h_q,r_q,Z\right)+\log\frac{P\left(\mathcal{G}_S,z|h_q,r_q,C_{r_q},\mathcal{G}_q\right)}{Q(Z)}\right]\\
&=\mathbb{E}_{Q(Z)}\left[\log P\left(t_q|h_q,r_q,Z\right)+\log\frac{P\left(z|C_{r_q}\right)}{Q(z)}+\log\frac{P\left(\mathcal{G}_S|h_q,r_q,z,\mathcal{G}_q\right)}{Q(\mathcal{G}_S)}\right]\\
&=\underbrace{\mathbb{E}_{Q(Z)}\left[\log P\left(t_q|h_q,r_q,Z\right)\right]}_{(1)}-\underbrace{KL\left(Q(z)\|P\left(z|C_{r_q}\right)\right)}_{(2)}\\
&-\underbrace{KL\left(Q\left(\mathcal{G}_S\right)\|P\left(\mathcal{G}_S|\mathcal{G}_q,z\right)\right)}_{(3)}.
\end{aligned}
\tag{A.5}
$$

Therefore, we obtain the unified objective function as presented in Eq. (22), enabling simultaneous optimization of the Neural Process-based Hypothesis Extractor and the Graph Stochastic Attention-based Predictor.

B ADDITIONAL ABLATION EXPERIMENTS

We have repeated the relevant experiments three times and reported both the mean and standard deviation of the results. As presented in Table 1, the obtained results demonstrate the robustness of our proposed method and clearly validate the contributions of each component.

Table 1. Ablation study (mean ± standard deviation over three runs) on NELL, ConceptNet, and WIKI datasets.

	NELL		**ConceptNet**		**WIKI**	
	MRR	**Hit@1**	**MRR**	**Hit@1**	**MRR**	**Hit@1**
GS-NP	**0.620±0.006**	**0.513±0.008**	**0.633±0.003**	**0.562±0.014**	**0.694±0.001**	**0.599±0.005**
w/o GSAT P	0.573±0.006	0.442±0.004	0.610±0.004	0.516±0.008	0.674±0.003	0.555±0.005
w/o NP E	0.592±0.005	0.468±0.009	0.614±0.005	0.534±0.024	0.687±0.003	0.586±0.005

REFERENCES

[1] Alexander A Alemi, Ian Fischer, Joshua V Dillon, and Kevin Murphy. 2016. Deep Variational Information Bottleneck. In *International Conference on Learning Representations*.

[2] Hareesh Bahuleyan, Lili Mou, Olga Vechtomova, and Pascal Poupart. 2017. Variational attention for sequence-to-sequence models. *arXiv preprint arXiv:1712.08207* (2017).

[3] Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. *Advances in Neural Information Processing Systems* 26 (2013).

[4] TANG Caifang, RAO Yuan, YU Hualei, SUN Ling, CHENG Jiamin, and WANG Yutian. 2021. Improving knowledge graph completion using soft rules and adversarial learning. *Chinese Journal of Electronics* 30, 4 (2021), 623–633.

[5] Mingyang Chen, Wen Zhang, Wei Zhang, Qiang Chen, and Huajun Chen. 2019. Meta Relational Learning for Few-Shot Link Prediction in Knowledge Graphs. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*. 4217–4226.

[6] Zhikai Chen, Haitao Mao, Hang Li, Wei Jin, Hongzhi Wen, Xiaochi Wei, Shuaiqiang Wang, Dawei Yin, Wenqi Fan, Hui Liu, et al. 2024. Exploring the potential of large language models (llms) in learning on graphs. *ACM SIGKDD Explorations Newsletter* 25, 2 (2024), 42–61.

[7] Haixuan Ding, Jialiang Tang, Sheng Wan, and Chen Gong. 2025. Exploring Neural Radiance Fields for Thermal View Synthesis Solely with Thermal Inputs. *Chinese Journal of Electronics* (2025).

[8] Manqing Dong, Feng Yuan, Lina Yao, Xiwei Xu, and Liming Zhu. 2020. Mamo: Memory-augmented meta-optimization for cold-start recommendation. In *Proceedings of the 26th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*. 688–697.

[9] Xinjie Fan, Shujian Zhang, Bo Chen, and Mingyuan Zhou. 2020. Bayesian attention modules. *Advances in Neural Information Processing Systems* 33 (2020), 16362–16376.

[10] Mikhail Galkin, Xinyu Yuan, Hesham Mostafa, Jian Tang, and Zhaocheng Zhu. 2023. Towards foundation models for knowledge graph reasoning. *arXiv preprint arXiv:2310.04562* (2023).

[11] Marta Garnelo, Dan Rosenbaum, Christopher Maddison, Tiago Ramalho, David Saxton, Murray Shanahan, Yee Whye Teh, Danilo Rezende, and SM Ali Eslami. 2018. Conditional neural processes. In *International Conference on Machine Learning*. PMLR, 1704–1713.

[12] Marta Garnelo, Jonathan Schwarz, Dan Rosenbaum, Fabio Viola, Danilo J Rezende, SM Eslami, and Yee Whye Teh. 2018. Neural processes. *arXiv preprint arXiv:1807.01622* (2018).

[13] Neil Zhenqiang Gong, Ameet Talwalkar, Lester Mackey, Ling Huang, Eui Chul Richard Shin, Emil Stefanov, Elaine Shi, and Dawn Song. 2014. Joint link prediction and attribute inference using a social-attribute network. *ACM Transactions on Intelligent Systems and Technology (TIST)* 5, 2 (2014), 1–20.

[14] Kun Guo, Wenzhong Guo, Enjie Ye, Yutong Fang, Jiachen Zheng, Ximeng Liu, and Kai Chen. 2023. Federated Clique Percolation for Privacy-preserving Overlapping Community Detection. *ACM Transactions on Intelligent Systems and Technology* 14, 4 (2023), 1–25.

[15] Timothy Hospedales, Antreas Antoniou, Paul Micaelli, and Amos Storkey. 2021. Meta-learning in neural networks: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 44, 9 (2021), 5149–5169.

[16] Yang Hu, Adriane Chapman, Guihua Wen, and Dame Wendy Hall. 2022. What can knowledge bring to machine learning?—a survey of low-shot learning for structured data. *ACM Transactions on Intelligent Systems and Technology (TIST)* 13, 3 (2022), 1–45.

[17] Qian Huang, Hongyu Ren, and Jure Leskovec. 2022. Few-shot relational reasoning via connection subgraph pretraining. *Advances in Neural Information Processing Systems* 35 (2022), 6397–6409.

[18] Dan Hunter. 1998. No Wilderness of Single Instances: Inductive Inference in Law. *Journal of Legal Education* 48, 3 (1998), 365–401.

[19] Eric Jang, Shixiang Gu, and Ben Poole. 2016. Categorical Reparameterization with Gumbel-Softmax. In *International Conference on Learning Representations*.

[20] Shaoxiong Ji, Shirui Pan, Erik Cambria, Pekka Marttinen, and Philip S. Yu. 2022. A Survey on Knowledge Graphs: Representation, Acquisition, and Applications. *IEEE Transactions on Neural Networks and Learning Systems* 33, 2 (2022), 494–514.

[21] Hyunjik Kim, Andriy Mnih, Jonathan Schwarz, Marta Garnelo, Ali Eslami, Dan Rosenbaum, Oriol Vinyals, and Yee Whye Teh. 2019. Attentive neural processes. *arXiv preprint arXiv:1901.05761* (2019).

[22] Jeongho Kim, Chanyeong Heo, and Jaehee Jung. 2025. ReCDAP: Relation-Based Conditional Diffusion with Attention Pooling for Few-Shot Knowledge Graph Completion. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 2848–2852.

[23] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

[24] Diederik P Kingma and Max Welling. 2013. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114* (2013).

[25] Huan Yee Koh, Anh TN Nguyen, Shirui Pan, Lauren T May, and Geoffrey I Webb. 2024. Physicochemical graph neural network for learning protein–ligand interaction fingerprints from sequence data. *Nature Machine Intelligence* 6, 6 (2024), 673–687.

[26] Huan Yee Koh, Yizhen Zheng, Madeleine Yang, Rohit Arora, Geoffrey I Webb, Shirui Pan, Li Li, and George M Church. 2025. AI-driven protein design. *Nature Reviews Bioengineering* (2025), 1–23.

[27] Huidong Liang and Junbin Gao. 2022. How Neural Processes Improve Graph Link Prediction. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing*. 3543–3547.

[28] Xixun Lin, Jia Wu, Chuan Zhou, Shirui Pan, Yanan Cao, and Bin Wang. 2021. Task-adaptive neural process for user cold-start recommendation. In *Proceedings of the Web Conference 2021*. 1306–1316.

[29] Jhih-Chen Liu, Chiao-Ting Chen, Chi Lee, and Szu-Hao Huang. 2024. Evolving knowledge graph representation learning with multiple attention strategies for citation recommendation system. *ACM Transactions on Intelligent Systems and Technology* 15, 2 (2024), 1–26.

[30] Yixin Liu, Kaize Ding, Huan Liu, and Shirui Pan. 2023. Good-d: On unsupervised graph out-of-distribution detection. In *Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining*. 339–347.

[31] Yixin Liu, Ming Jin, Shirui Pan, Chuan Zhou, Yu Zheng, Feng Xia, and Philip S Yu. 2023. Graph Self-Supervised Learning: A Survey. *IEEE Transactions on Knowledge & Data Engineering* 35, 6 (2023), 5879–5900.
[32] Linhao Luo, Yixiang Fang, Xin Cao, Xiaofeng Zhang, and Wenjie Zhang. 2021. Detecting communities from heterogeneous graphs: A context path-based graph neural network model. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*. 1170–1180.
[33] Linhao Luo, Yixiang Fang, Moli Lu, Xin Cao, Xiaofeng Zhang, and Wenjie Zhang. 2023. GSim: A Graph Neural Network Based Relevance Measure for Heterogeneous Graphs. *IEEE Transactions on Knowledge & Data Engineering* 01 (2023), 1–14.
[34] Linhao Luo, Gholamreza Haffari, and Shirui Pan. 2023. Graph sequential neural ode process for link prediction on dynamic and sparse graphs. In *Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining*. 778–786.
[35] Linhao Luo, Yuan Fang Li, Reza Haf, and Shirui Pan. 2024. Reasoning on Graphs: Faithful and Interpretable Large Language Model Reasoning. In *International Conference on Learning Representations*.
[36] Linhao Luo, Yuan-Fang Li, Gholamreza Haffari, and Shirui Pan. 2023. Normalizing flow-based neural process for few-shot knowledge graph completion. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 900–910.
[37] Linhao Luo, Zicheng Zhao, Gholamreza Haffari, Yuan-Fang Li, Chen Gong, and Shirui Pan. 2025. Graph-constrained Reasoning: Faithful Reasoning on Knowledge Graphs with Large Language Models. In *Forty-second International Conference on Machine Learning*. PMLR.
[38] Linhao Luo, Zicheng Zhao, Gholamreza Haffari, Dinh Phung, Chen Gong, and Shirui Pan. 2025. GFM-RAG: graph foundation model for retrieval augmented generation. *arXiv preprint arXiv:2502.01113* (2025).
[39] Xin Lv, Yuxian Gu, Xu Han, Lei Hou, Juanzi Li, and Zhiyuan Liu. 2019. Adapting meta knowledge graph information for multi-hop reasoning over few-shot relations. *arXiv preprint arXiv:1908.11513* (2019).
[40] Siqi Miao, Mia Liu, and Pan Li. 2022. Interpretable and generalizable graph learning via stochastic attention mechanism. In *International Conference on Machine Learning*. PMLR, 15524–15543.
[41] Tom Mitchell, William Cohen, Estevam Hruschka, Partha Talukdar, Bishan Yang, Justin Betteridge, Andrew Carlson, Bhavana Dalvi, Matt Gardner, and Bryan Kisiel. 2018. Never-ending learning. *Commun. ACM* 61, 5 (2018), 103–115.
[42] Guanglin Niu, Yang Li, Chengguang Tang, Ruiying Geng, Jian Dai, Qiao Liu, Hao Wang, Jian Sun, Fei Huang, and Luo Si. 2021. Relational learning with gated and attentive neighbor aggregator for few-shot knowledge graph completion. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 213–222.
[43] Haoran Niu, Haitao He, Jianzhou Feng, Junlan Nie, Yangsen Zhang, and Jiadong Ren. 2022. Knowledge graph completion based on GCN of multi-information fusion and high-dimensional structure analysis weight. *Chinese Journal of Electronics* 31, 2 (2022), 387–396.
[44] Shirui Pan, Linhao Luo, Yufei Wang, Chen Chen, Jiapu Wang, and Xindong Wu. 2024. Unifying Large Language Models and Knowledge Graphs: A Roadmap. *IEEE Transactions on Knowledge & Data Engineering* 01 (2024), 1–20.
[45] Shirui Pan, Yizhen Zheng, and Yixin Liu. 2024. Integrating graphs with large language models: Methods and prospects. *IEEE Intelligent Systems* 39, 1 (2024), 64–68.
[46] Zhangchi Qiu, Ye Tao, Shirui Pan, and Alan Wee-Chung Liew. 2024. Knowledge Graphs and Pretrained Language Models Enhanced Representation Learning for Conversational Recommender Systems. *IEEE Transactions on Neural Networks and Learning Systems* (2024), 1–15.
[47] Liu Ran, Zhongzhou Liu, Xiaoli Li, and Yuan Fang. 2024. Context-Aware Adapter Tuning for Few-Shot Relation Learning in Knowledge Graphs. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*. 17525–17537.
[48] Xuhui Ren, Tong Chen, Quoc Viet Hung Nguyen, Lizhen Cui, Zi Huang, and Hongzhi Yin. 2024. Explicit knowledge graph reasoning for conversational recommendation. *ACM Transactions on Intelligent Systems and Technology* 15, 4 (2024), 1–21.
[49] Shiv Shankar and Sunita Sarawagi. 2019. Posterior attention models for sequence to sequence learning. In *International Conference on Learning Representations*.
[50] Jiawei Sheng, Shu Guo, Zhenyu Chen, Juwei Yue, Lihong Wang, Tingwen Liu, and Hongbo Xu. 2020. Adaptive Attentional Network for Few-Shot Knowledge Graph Completion. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*. 1681–1691.
[51] Gautam Singh, Jaesik Yoon, Youngsung Son, and Sungjin Ahn. 2019. Sequential neural processes. *Advances in Neural Information Processing Systems* (2019), 10254–10264.
[52] Robyn Speer, Joshua Chin, and Catherine Havasi. 2017. Conceptnet 5.5: An open multilingual graph of general knowledge. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 31.
[53] Komal Teru, Etienne Denis, and Will Hamilton. 2020. Inductive relation prediction by subgraph reasoning. In *International Conference on Machine Learning*. PMLR, 9448–9457.
[54] Naftali Tishby and Noga Zaslavsky. 2015. Deep learning and the information bottleneck principle. In *2015 IEEE Information Theory Workshop*. IEEE, 1–5.
[55] Nicolaas G Van Kampen. 1976. Stochastic differential equations. *Physics Reports* 24, 3 (1976), 171–228.
[56] Denny Vrandečić and Markus Krötzsch. 2014. Wikidata: a free collaborative knowledgebase. *Commun. ACM* 57, 10 (2014), 78–85.

[57] Guojia Wan, Shirui Pan, Chen Gong, Chuan Zhou, and Gholamreza Haffari. 2021. Reasoning like human: hierarchical reinforcement learning for knowledge graph reasoning. In *Proceedings of the Twenty-Ninth International Conference on International Joint Conferences on Artificial Intelligence*. 1926–1932.
[58] Sheng Wan, Yibing Zhan, Shuo Chen, Shirui Pan, Jian Yang, Dacheng Tao, and Chen Gong. 2023. Boosting Graph Contrastive Learning via Adaptive Sampling. *IEEE Transactions on Neural Networks and Learning Systems* (2023), 1–13.
[59] Hongwei Wang, Hongyu Ren, and Jure Leskovec. 2021. Relational message passing for knowledge graph completion. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*. 1697–1707.
[60] Siheng Xiong, Ali Payani, Ramana Kompella, and Faramarz Fekri. 2024. Large Language Models Can Learn Temporal Reasoning. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics*. 10452–10470.
[61] Siheng Xiong, Yuan Yang, Ali Payani, James C Kerce, and Faramarz Fekri. 2024. Teilp: Time prediction over knowledge graphs via logical reasoning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 38. 16112–16119.
[62] Wenhan Xiong, Mo Yu, Shiyu Chang, Xiaoxiao Guo, and William Yang Wang. 2018. One-Shot Relational Learning for Knowledge Graphs. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. 1980–1990.
[63] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. 2015. Show, attend and tell: Neural image caption generation with visual attention. In *International Conference on Machine Learning*. PMLR, 2048–2057.
[64] Xiaohan Xu, Peng Zhang, Yongquan He, Chengpeng Chao, and Chaoyang Yan. 2022. Subgraph neighboring relations infomax for inductive link prediction on knowledge graphs. *arXiv preprint arXiv:2208.00850* (2022).
[65] Chuxu Zhang, Huaxiu Yao, Chao Huang, Meng Jiang, Zhenhui Li, and Nitesh V Chawla. 2020. Few-shot knowledge graph completion. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34. 3041–3048.
[66] Zicheng Zhao, Linhao Luo, Shirui Pan, Quoc Viet Hung Nguyen, and Chen Gong. 2023. Towards Few-shot Inductive Link Prediction on Knowledge Graphs: A Relational Anonymous Walk-guided Neural Process Approach. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 515–532.
[67] Yizhen Zheng, Huan Yee Koh, Jiaxin Ju, Anh TN Nguyen, Lauren T May, Geoffrey I Webb, and Shirui Pan. 2025. Large language models for scientific discovery in molecular property prediction. *Nature Machine Intelligence* (2025), 1–11.

Received 7 April 2025; revised 8 October 2025; accepted 16 October 2025