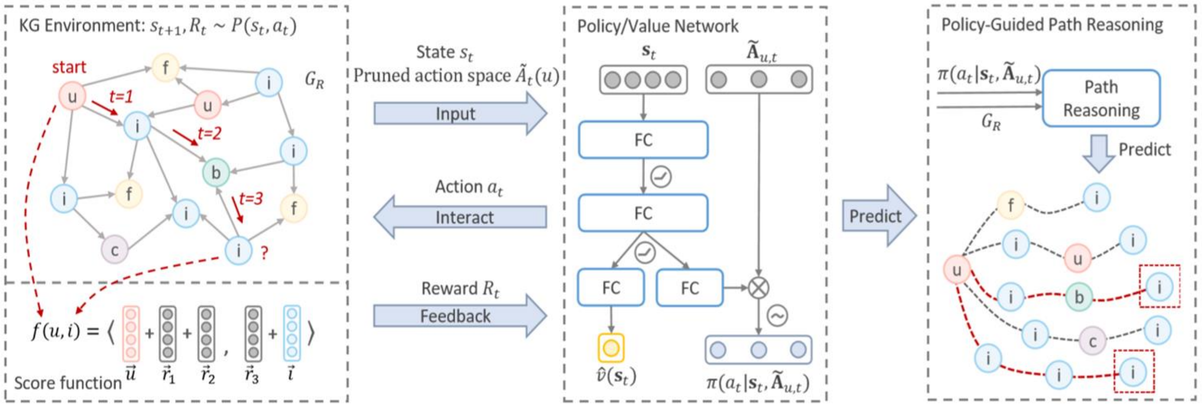
Basic Pipeline Structure for PGPR

The algorithm aims to learn a policy that navigates from a user to potential items of interest by interacting with the knowledge graph environment. The trained policy is then adopted for the path reasoning phase to make recommendations to the user.



KG Environment: $s_{t+1}, R_t \sim P(s_t, a_t)$
start
G_R
t=1
t=2
t=3
?
u
i
f
b
c
$f(u,i) = \langle \vec{u} + \vec{r}_1 + \vec{r}_2 , \vec{r}_3 + \vec{i} \rangle$
Score function
State s_t
Pruned action space $\tilde{A}_t(u)$
Input
Action a_t
Interact
Reward R_t
Feedback
Policy/Value Network
$\mathbf{s}_t$
$\tilde{\mathbf{A}}_{u,t}$
FC
FC
FC
FC
$\hat{v}(\mathbf{s}_t)$
$\pi(a_t|\mathbf{s}_t, \tilde{\mathbf{A}}_{u,t})$
Predict
Policy-Guided Path Reasoning
$\pi(a_t|\mathbf{s}_t, \tilde{\mathbf{A}}_{u,t})$
G_R
Path Reasoning
Predict

It solves the problem through reinforcement learning by making recommendations while simultaneously searching for paths in the context of rich heterogeneous information in the KG.

Given a user u , the goal is to find a set of candidate items {in} and the corresponding reasoning paths {p_n(u,i_n)}.One method is to just have absolute rewards and sample n paths using the independent probabilities of all the items given the state and action space i.e. π(·|s, A˜ u) . But, the problem with this is that agent might lead to a path with more rewards and not explore the other ones.

This leads to the overfitting problem as the parameters are now used only for some paths. So, employ beam search guided by the action probability and reward to explore the candidate paths as well as the recommended items for each user. We forcefully reset some probabilities to avoid the condition in which the agent moves on the same path again and again.

We do this by the following algorithm:

It takes as input the given user u, the policy network π(·|s, A˜ u), horizonT , and predefined sampling sizes at each step, denoted by K_1, . . . ,K_T . As output, it delivers a candidate set of T -hop paths P_T for the user with corresponding path generative probabilities Q_T and path rewards R_T . Note that each path p_T (u,i_n) ∈ P_T ends with an item entity associated with a path generative probability and a path reward.

Thus, for each pair of (u,i_n) in the candidate set, we select the path from P_T with the highest generative probability based on Q_T as the one to interpret the reasoning process of why item in is recommended to u. Finally, we rank the selected interpretable paths according to the path reward in R_T and recommend the corresponding items to the user.