A Appendix

A.1 List of abbreviations

- **AI** Artificial Intelligence
- **AMDP** Abstract Markov Decision Process
- **API** Application Programming Interface
- **DQN** Deep Q-Network
- **ER** Entity and Relation (linking)
- **KG** Knowledge Graph
- **KGE** Knowledge Graph Embedding
- **KGEM** Knowledge Graph Embedding Model
- **KGRL** Knowledge Graphs (Injection in) Reinforcement Learning
- **KNN** K-Nearest Neighbors algorithm
- **LLM** Large Language Model
- **LM** Language Model
- **MDP** Markov Decision Process
- **ML** Machine Learning
- **NLU** Natural Language Understanding
- **NLP** Natural Language Processing
- **OWL** Web Ontology Language
- **RDF** Resource Description Framework
- **RL** Reinforcement Learning
- **RLHF** Reinforcement Learning from Human Feedback
- **RW** Random Walk
- **SEM** The Simple Event Model
- **W3C** WWW Consortium

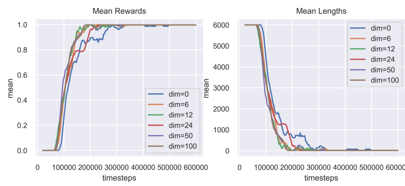
A.2 Additional Learning curves

A.3 Ablation studies

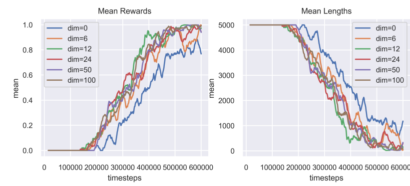
Dimesnions of KGE We test the performance of different dimensions of the KGE, by evaluating

A.4 Hyperparameters tuning

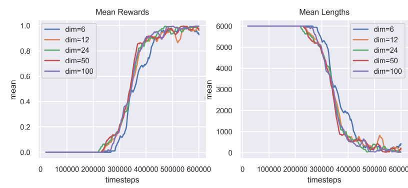
We perform Hyperparameter tuning to obtain good training parameters for the Baseline approach (DQN), in order to ensure the possibility of convergence for the Baseline. We do not perform Hyperparameter tuning for the parameters of the proposed methods even though further performance incereases would be possible.



(a) Dimensions - DoorKey



(b) Dimensions - KeyKorridor



(c) Dimensions - ObstructedMaze

Fig. 1. Performance of different dimensions for the KGE model.

A.5 Training Runs with Random Action

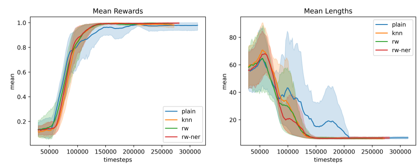
To test the generalization of the KGRL approach, we additionally evaluated the trained policy with introducing some noise. To this, end 10 episodes are completed at each evaluation step in which the trained policy is perturbed by a random action with small probability to verify that the policy has not been overfitted. The obtained results support the claim that KG injection into learning process improves sample efficiency, esspecially for complex environments

B Complexity of Minigrid Environments

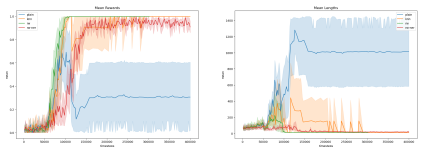
Table B describes the distribution of environments across 3 complexity classes (easy, medium and hard) according to the complexity of the knowledge graph, describing this environment.

Environment	N, nodes	N, edges	N, artifacts	Obstacles	Complexity
Lava Gap s5	84	140	0	yes	Easy
Lava Gap s7	156	288	0	yes	Medium
Door Key 5x5	86	132	1-2	no	Easy
Door Key 8x8	155	319	1-2	no	Medium
Lava Crossing	253	492	0	yes	Hard
Key Corridor s3r2	116	182	1-2	no	Hard
Obstructed Maze D1	775	1465	3+	yes	Hard

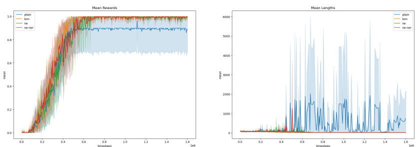
Table 1. Distribution of Minigrid Environments into Complexity Classes



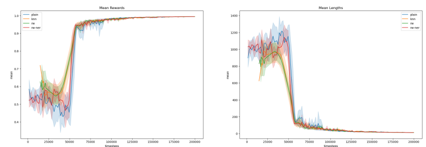
(a) Lava Gap s5



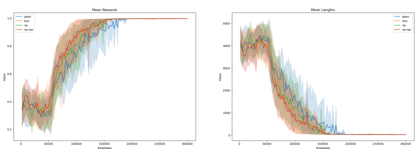
(b) Lava Gap s7



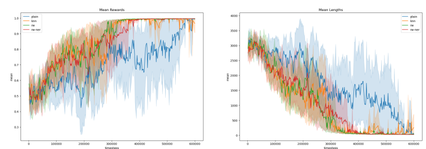
(c) Lava Crossing



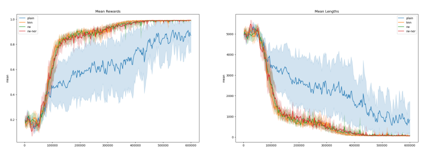
(d) Door Key 5x5



(e) Door Key 8x8



(f) Key Korridor s3r2



(g) Obstructed Maze D1

Fig. 2. Mean Reward and Episode length achieved by the trained approaches for different environments. Rewards and episode length are reported as average over ten seeds with one standard deviation confidence intervals. We evaluated the policies at 200 equally distance training steps running ten episodes at every evaluation with a small probability of disturbing the agent with a random action to test for overfitting.

B.1 Classes of the Minigrid Ontology

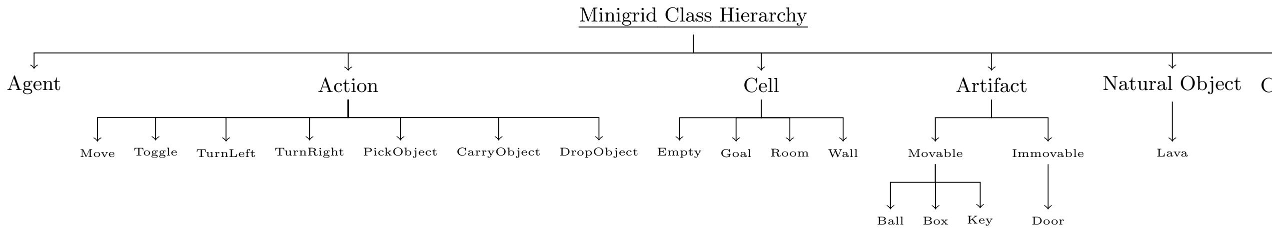


Fig. 3. The Minigrid class hierarchy in the proposed Ontology

C Domain Adaptation Process

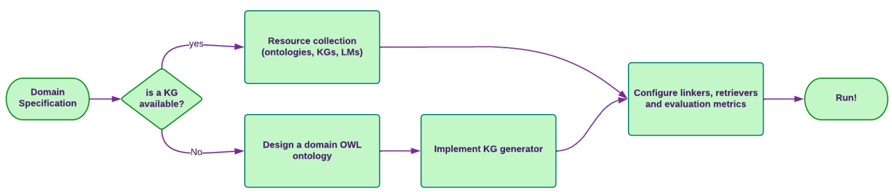


Fig. 4. Tailoring KGRL pipeline to a new domain: The image describes steps and data, that are required to adapt KGRL pipeline to a new domain. Resources (knowledge graphs, ontologies, language models, multimodal embeddings, etc.) reusability plays a major role in this process.

link to anonymous gitlab TODO: fix ontology uri names -¿ labels