Learning Local Forward Models on Unforgiving Games

by Alexander Dockhorn, Simon Lucas, Vanessa Volz, Ivan Bravi,
Raluca Gaina and Diego Perez Liebana

Institute for Intelligent Cooperating Systems
Department for Computer Science, Otto von Guericke University Magdeburg
Universitätsplatz 2, 39106 Magdeburg, Germany

Email: alexander.dockhorn@ovgu.de

Contents

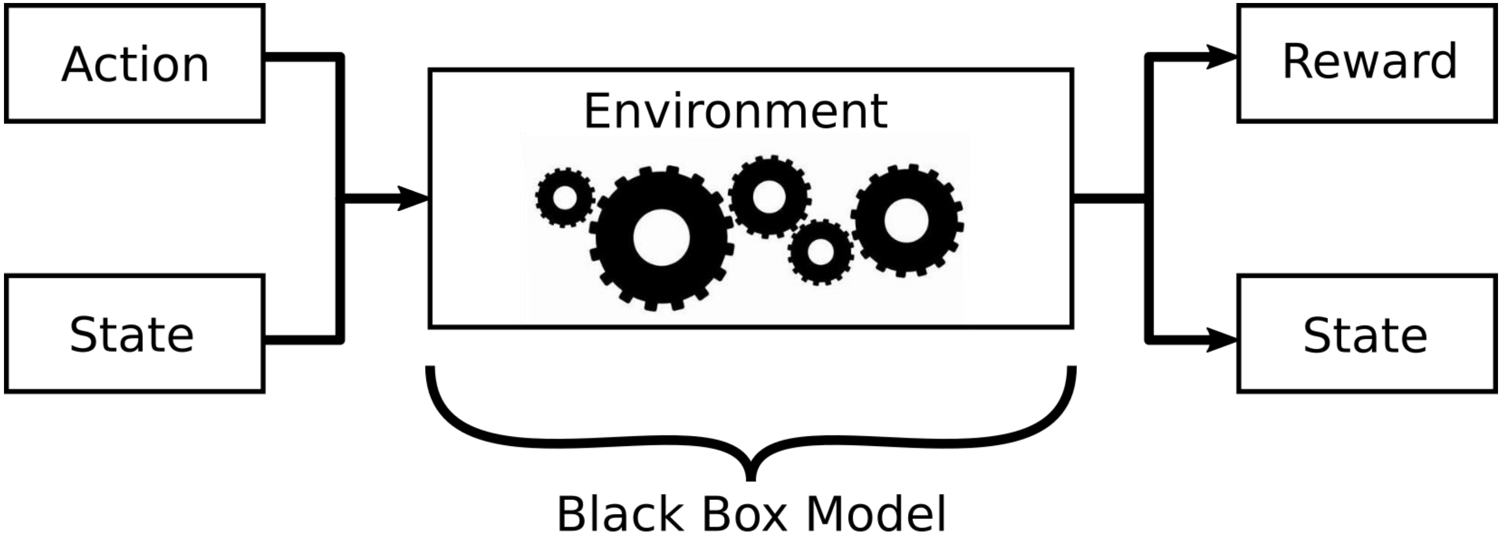
I. Forward Model Learning

II. Sokoban – An Unforgiving Game

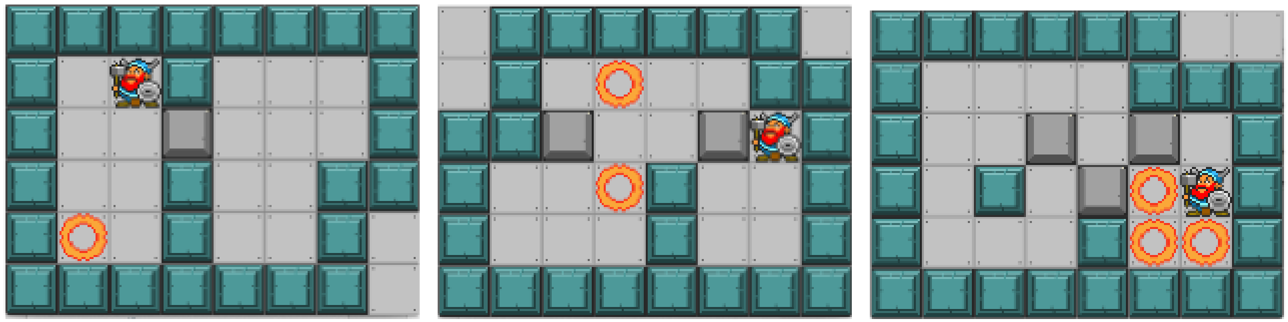
III. Local Forward Models

IV. Summary and Future Challenges

Forward Model Learning



Sokoban – An Unforgiving Game

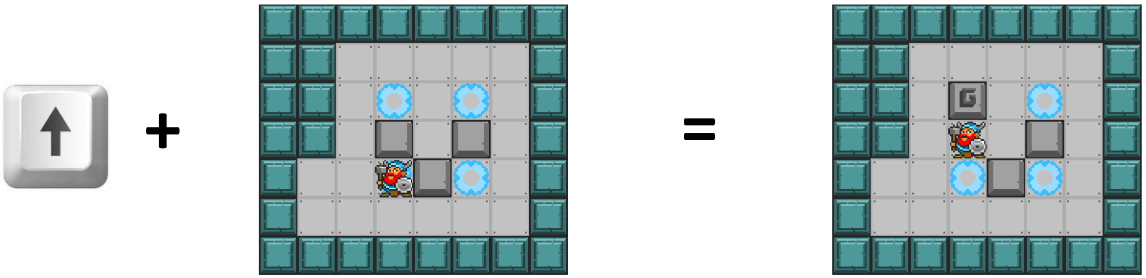


- Sokoban:
  - The avatar needs to push the stones onto the circles
  - Stones can only be pushed in case the next tile is empty
  - If all circles are covered by a stone the player wins

- Single mistake can lead to a overall failure
  - e.g. pushing the stone into a corner

End-to-End Forward Models

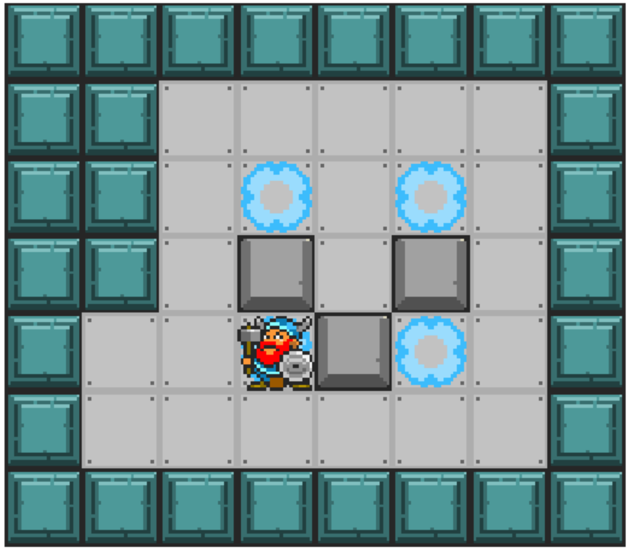
- Predict the next state based on the current state and action



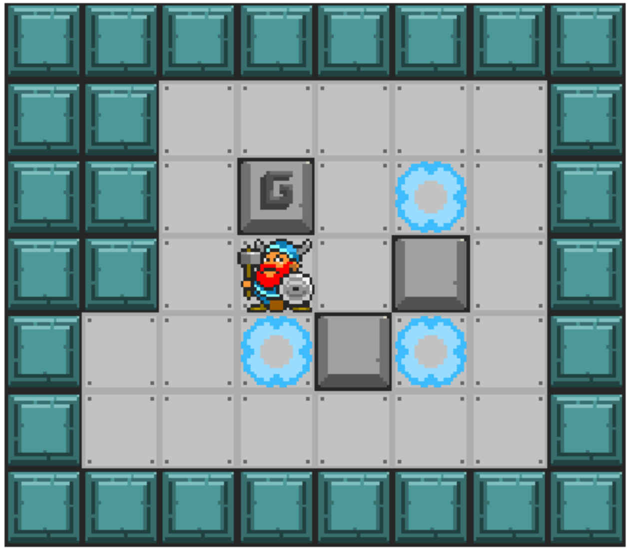
- Problems:
  - Many training examples necessary
  - Each game tick provides only one example
  - The resulting state-transition function is overly complex

Local Forward Models - Motivation

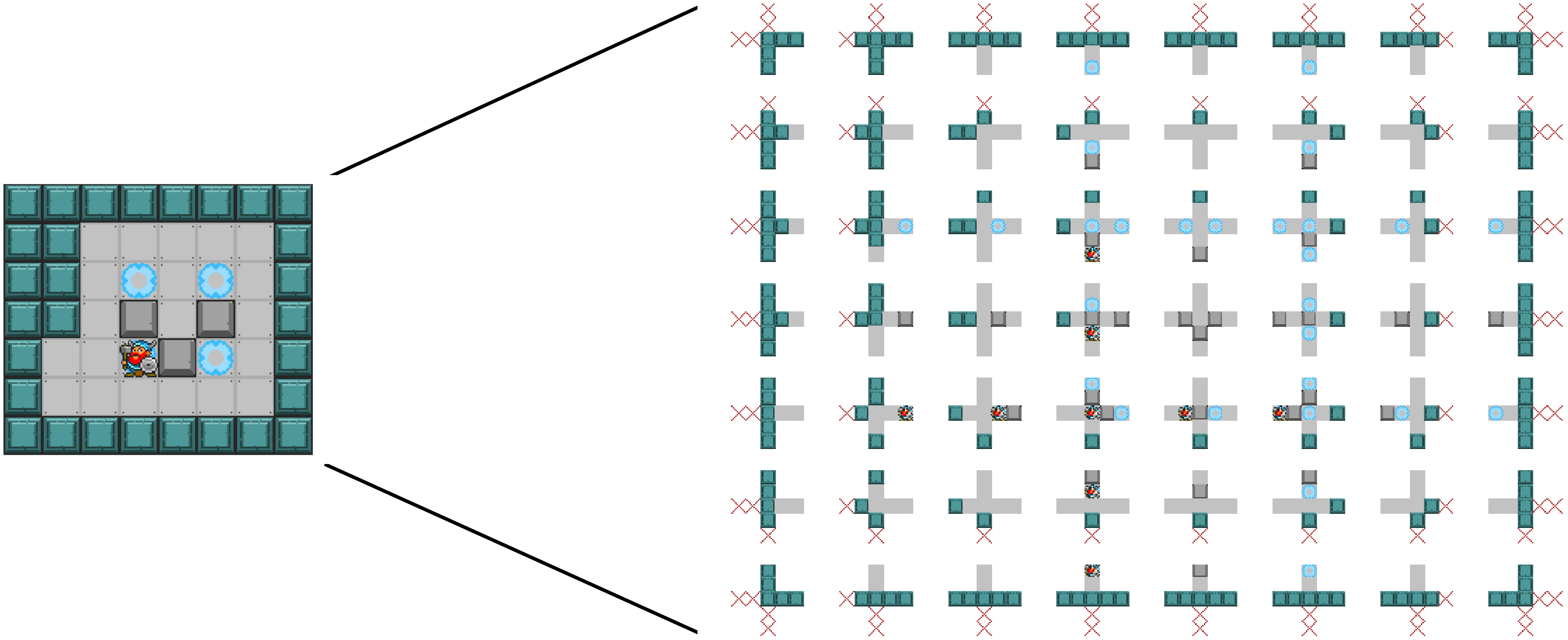
Current State



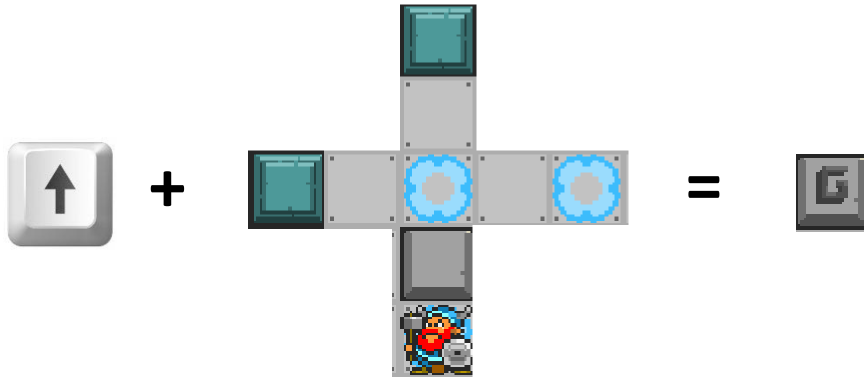
Next State

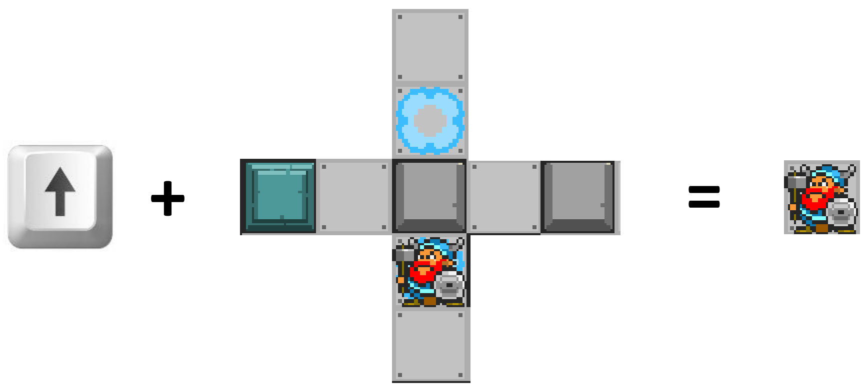


Local Forward Models

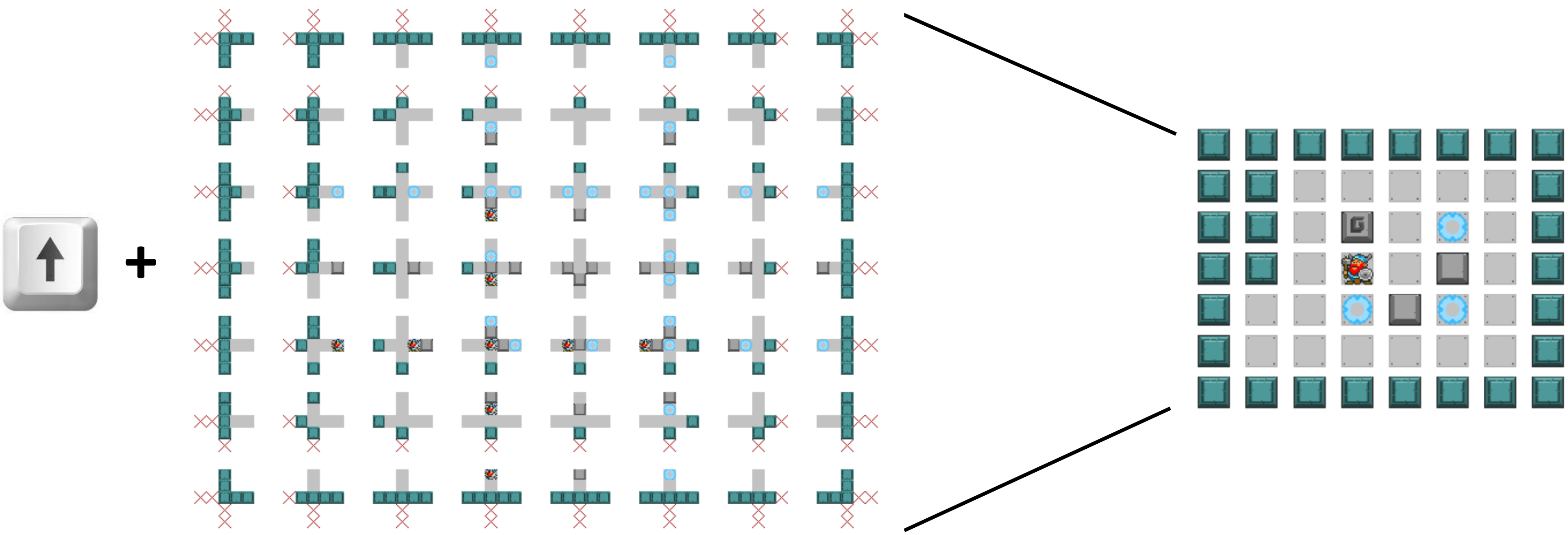


Local Forward Models





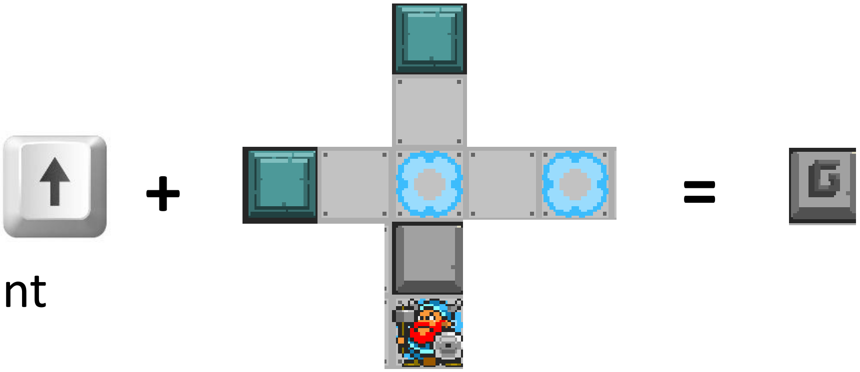
Local Forward Models



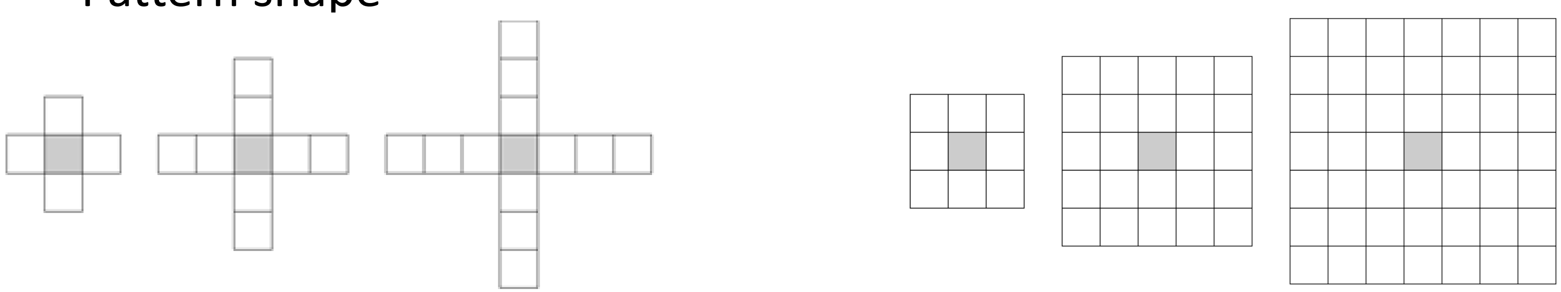
Local Forward Models - Implementation

- Each pattern provides a training example for any supervised learning method

- In this work, we used:
  - a simple hash map
  - and a decision tree
  - In combination with a RHEA agent



- We studied the influence of:
  - Pattern size
  - Pattern shape



Local Forward Models - Results

- Best results were achieved when using the cross pattern with a span of 2
  - Which is matching the true model

- Decision trees are limited in their generalizability, but other classifiers, e.g. DL, may yield much better results

- The learned model transfers well to previously unseen levels.

	span	unique patterns	easy acc	easy score	hard acc	hard score
cross	1	5000	0.9930	0.73	0.9965	0.64
	2	27419	0.9799	0.42	0.9894	0.65
	3	46271	0.9771	0.41	0.9869	0.65
square	1	27667	0.9822	0.35	0.9919	0.65
	2	151995	0.9773	0.43	0.9864	0.64
	3	303200	0.9770	0.46	0.9863	0.65

TABLE II: Results of the *Hash Map Model*

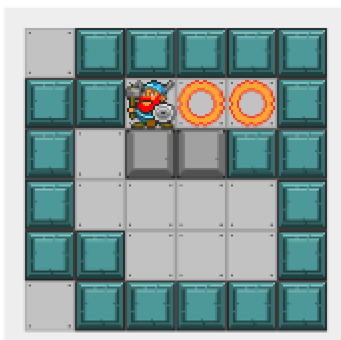
	span	nr of tree nodes	easy acc	easy score	hard acc	hard score
cross	1	679	0.9959	0.00	0.9973	0.67
	2	459	0.9991	1.11	0.9997	0.80
	3	664	0.9990	0.75	0.9995	0.78
square	1	1638	0.9972	0.00	0.9975	0.63
	2	2088	0.9975	1.06	0.9988	0.63
	3	2864	0.9981	0.66	0.9985	0.67

TABLE III: Results of the *Decision Tree Model*

Observed Problems I

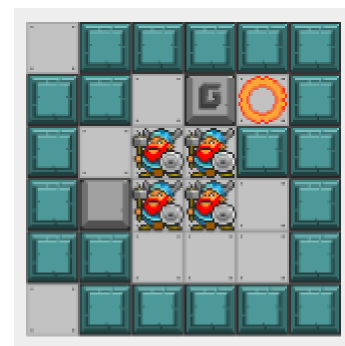
- A single error during the prediction can spread fast

Start State



Final State

Predicted State



Observed Problems II

- All resulting models had a accuracy above 0.98, hower assuming a static environment yields similar accuracy
  - What other measures can be used to rate the applicability of the learned model?
  - How to measure the confidence or risk of our model?

- Choosing the pattern span and pattern shape is an inherent feature selection problem.
  - Can this be solved at run-time?

Summary

- Local Forward Models can be efficient in learning forward models
  - Due to the convolution many training samples are produced for a single game tick
  - They can be easily transferred to unseen levels

- **Next challenges:**
  - Adapt local forward models to more complex problems (e.g. GVGAI)
  - Rating the model's precision or the agent's risk at run-time
  - Learning local forward models for non-markovian games

Thank you for your attention!

by Alexander Dockhorn, Simon Lucas, Vanessa Volz, Ivan Bravi,
Raluca Gaina and Diego Perez Liebana

Institute for Intelligent Cooperating Systems
Department for Computer Science, Otto von Guericke University Magdeburg
Universitätsplatz 2, 39106 Magdeburg, Germany

Email: alexander.dockhorn@ovgu.de