Learning Discrete World Models

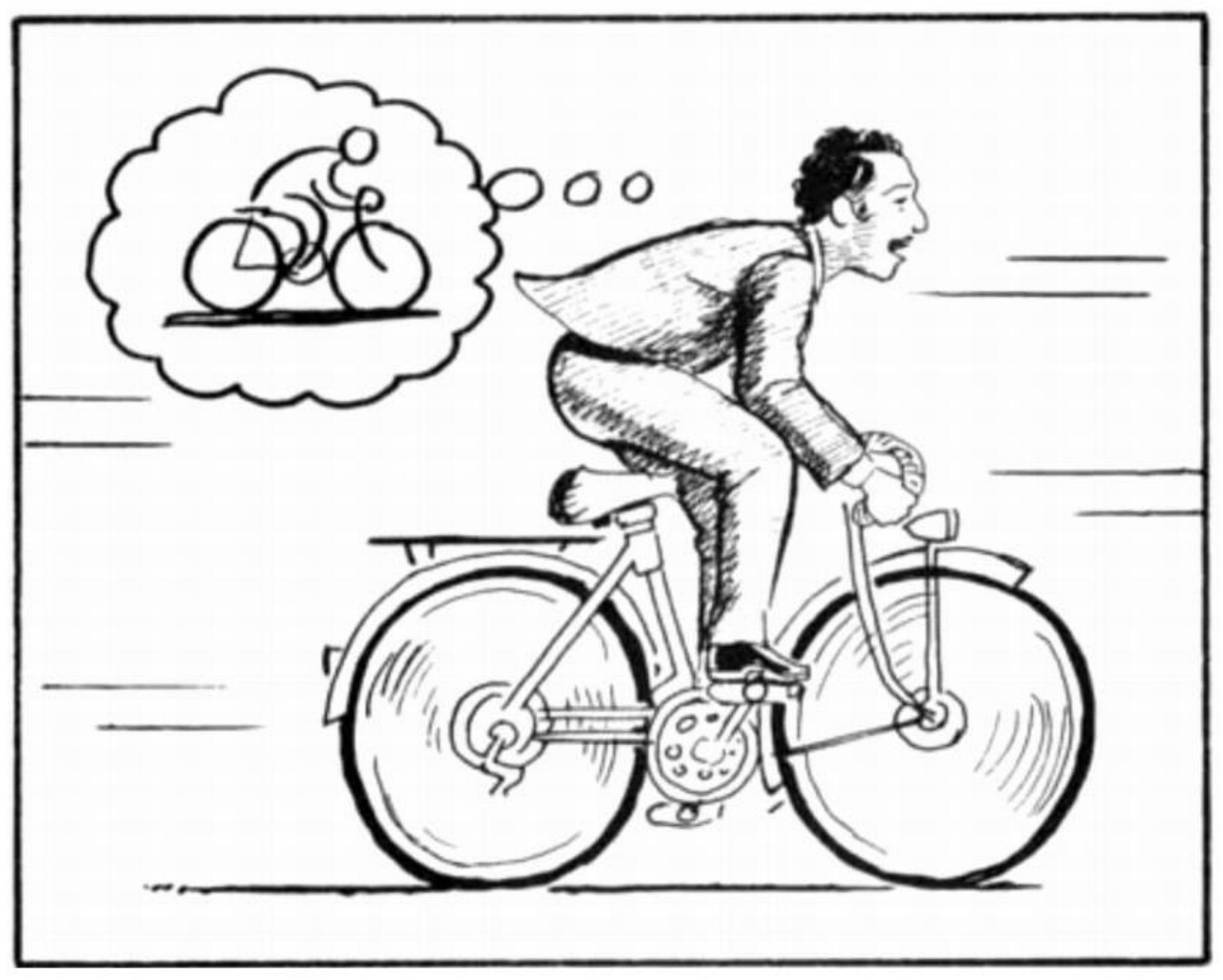
Bruce Brasseur, Dr. Pooyan Jamshidi, Dr. Forest Agostinelli
University of South Carolina
Dept. of Computer Science and Engineering

Abstract

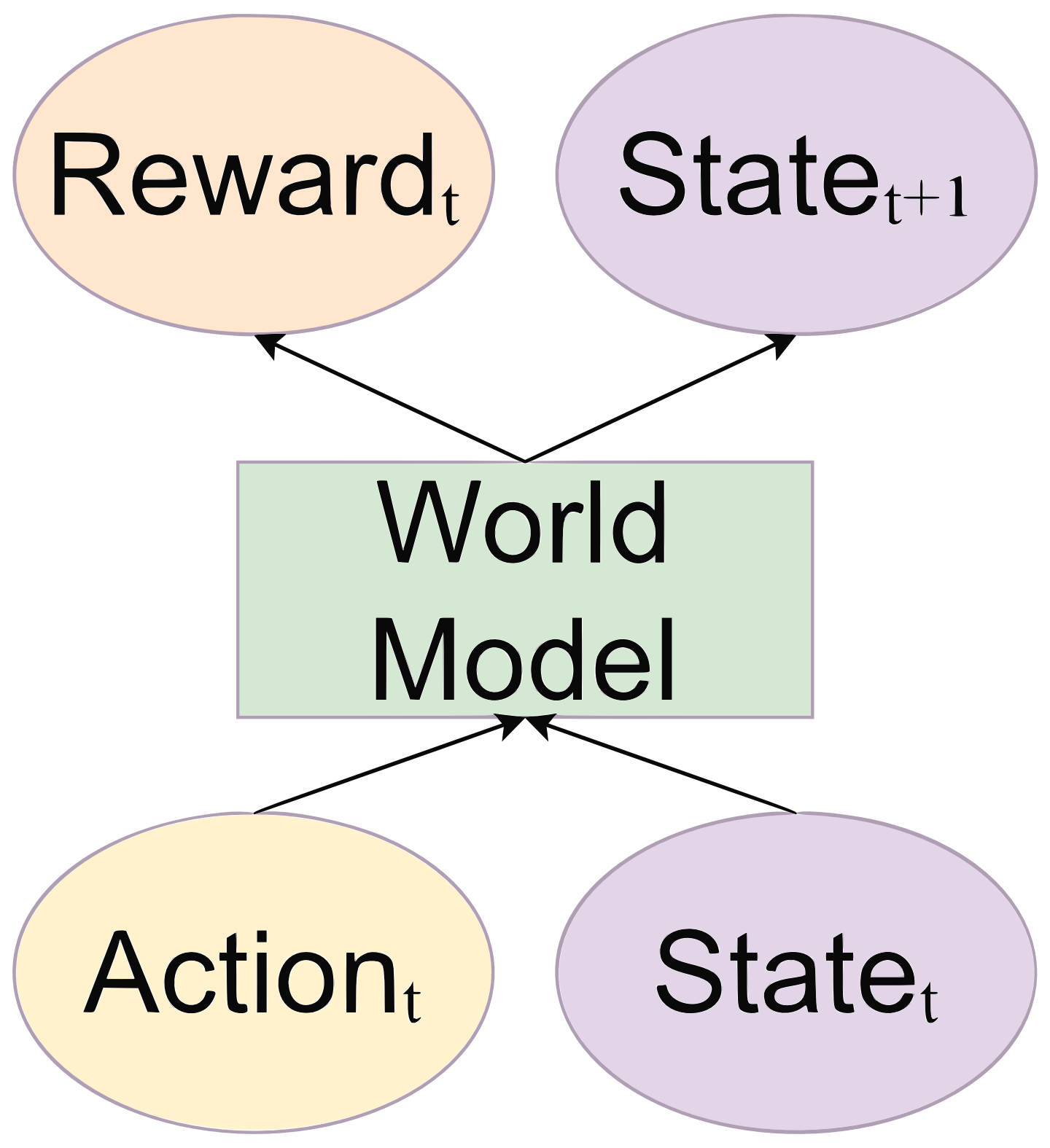
The incorporation of world models into reinforcement learning architectures has in many cases shown to improve performance, sample efficiency, and robustness, but implementations focusing on continuous control tasks may still be liable to **model degradation**. We propose that environment data be discretized in an attempt to reduce/eliminate model degradation.

World Models

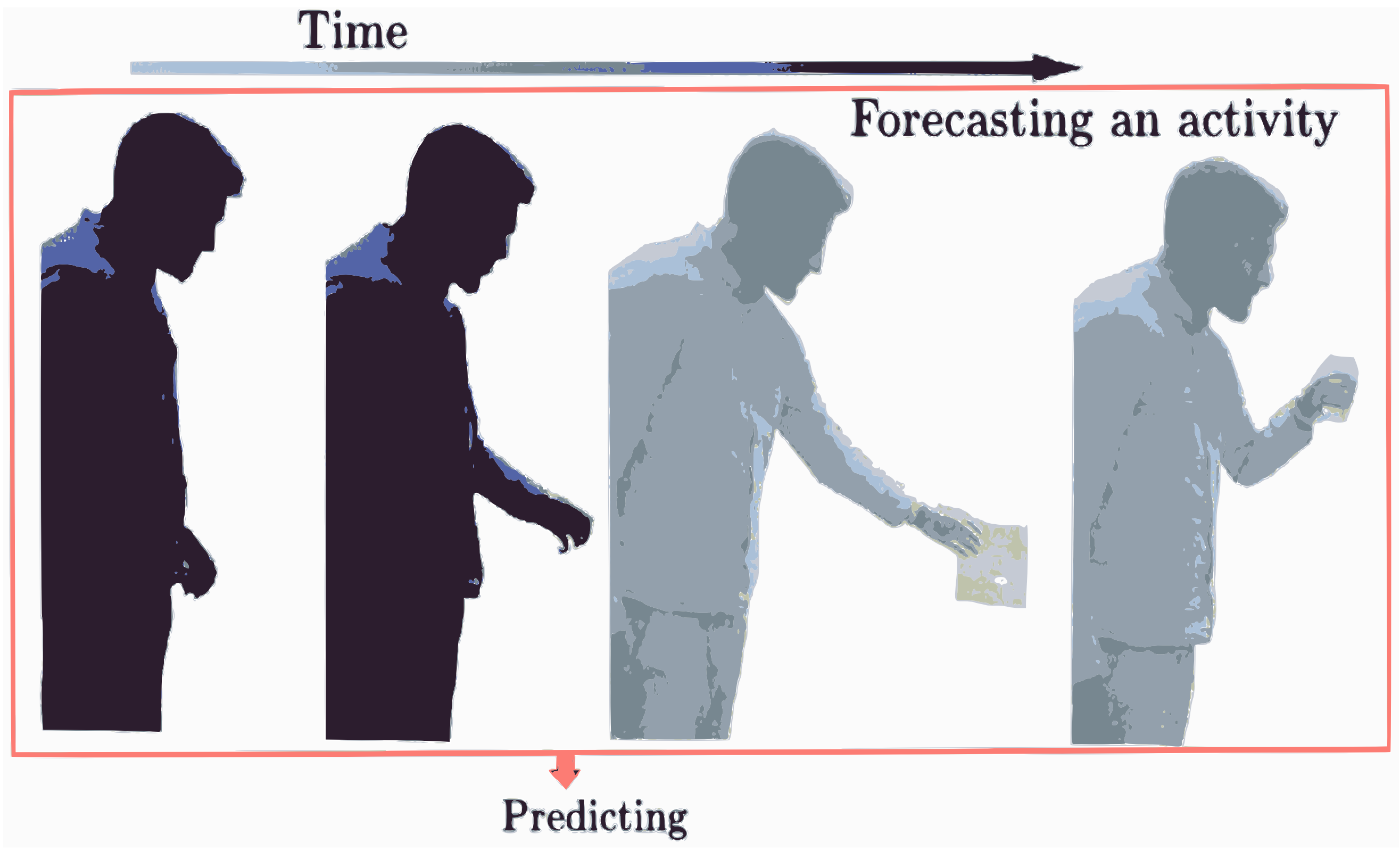
Before learning any policies, agents are left to explore their environment in an **unsupervised** manner. A recursive network learns to predict future environment states given past one's + actions taken.



Once the world model is sufficiently trained, a policy network can train using it instead of the actual environment.

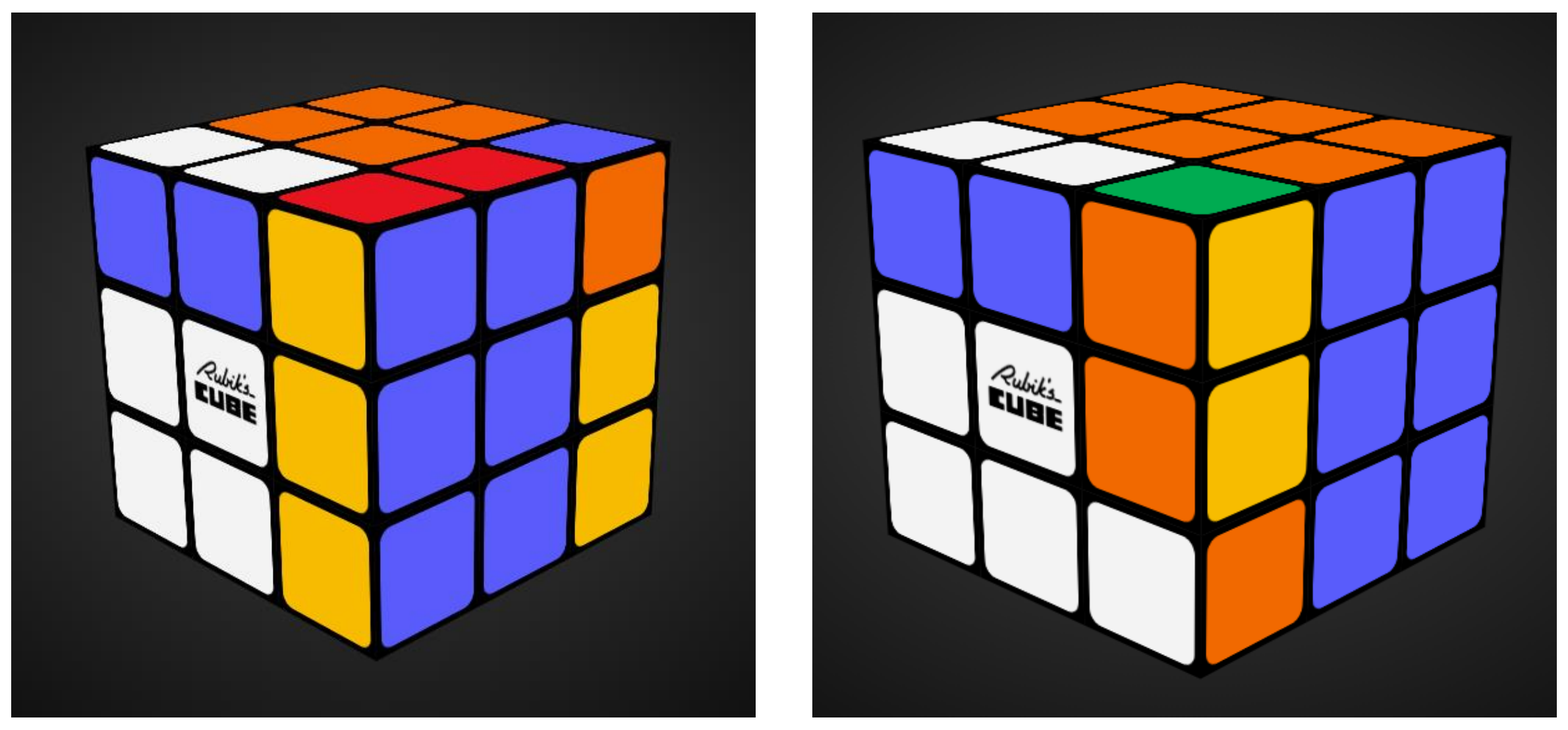


Trained world model can also guide decision making by predicting potential consequences in the real environment during run time.

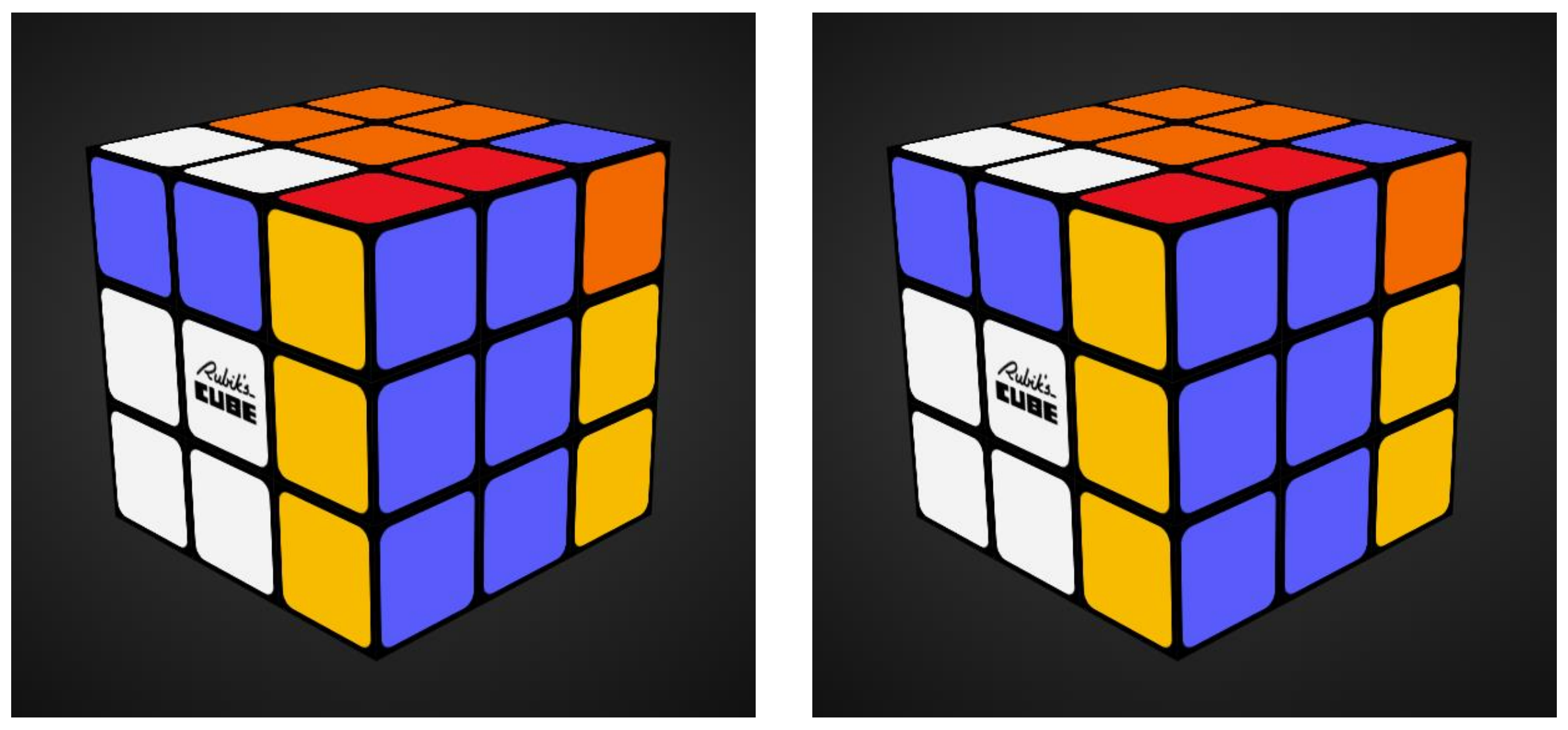


Model Degradation

Slightly suboptimal floating point values in a world model **can propagate into noticeable differences between world model and real environment.**



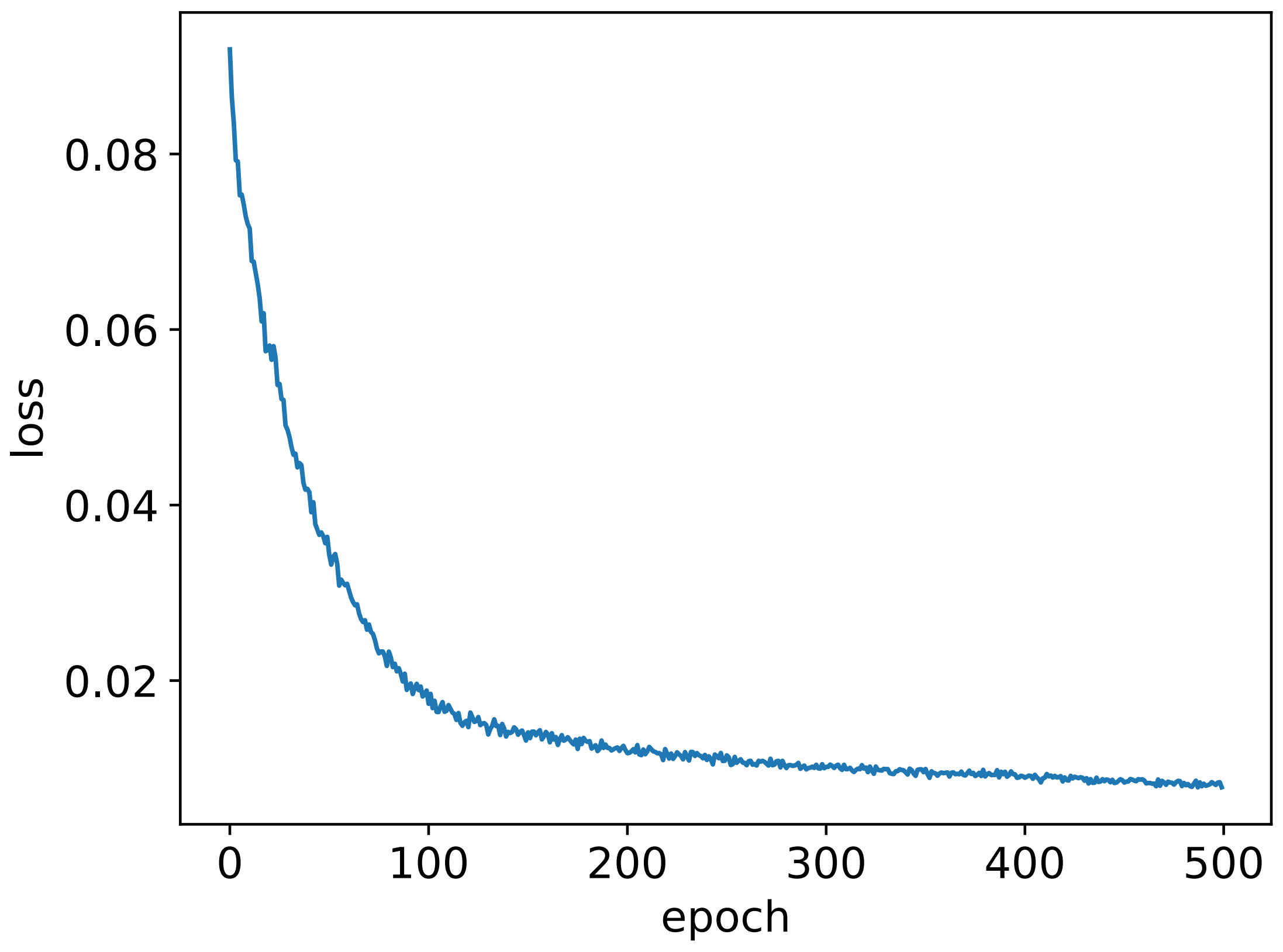
Predicted state of Rubik's cube after the moves R, U R' vs actual state



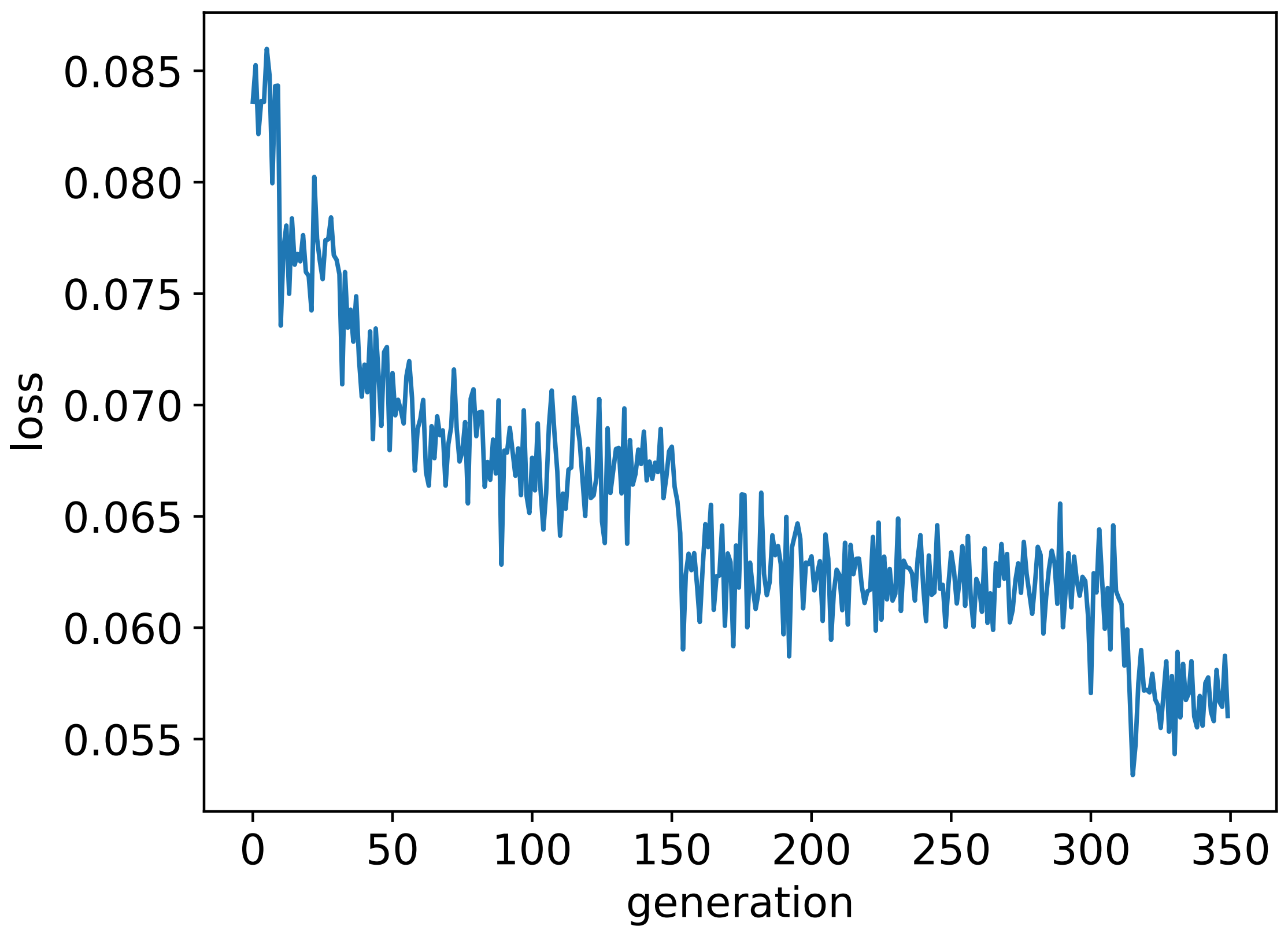
Thresholding environment data to 1 or 0 eliminates error accumulation over 3 moves and results in matching configurations.

Methods

Due to discretization functions having a slope of 0, traditional gradient based methods are unavailable to us. We therefore ran tests with straight through estimators as well as evolutionary methods.



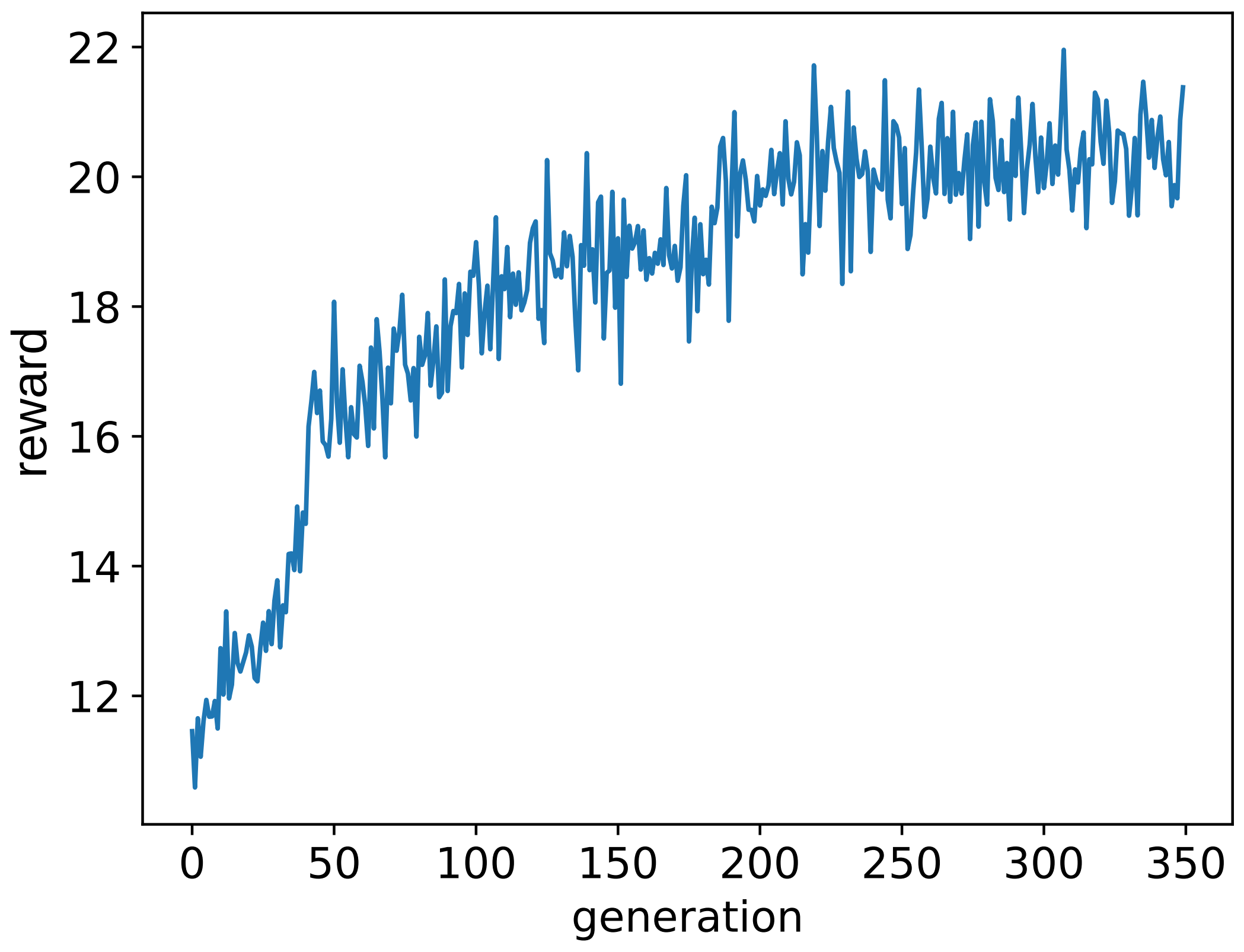
Straight through estimators provided much faster training, but can have trouble exploring large search spaces.

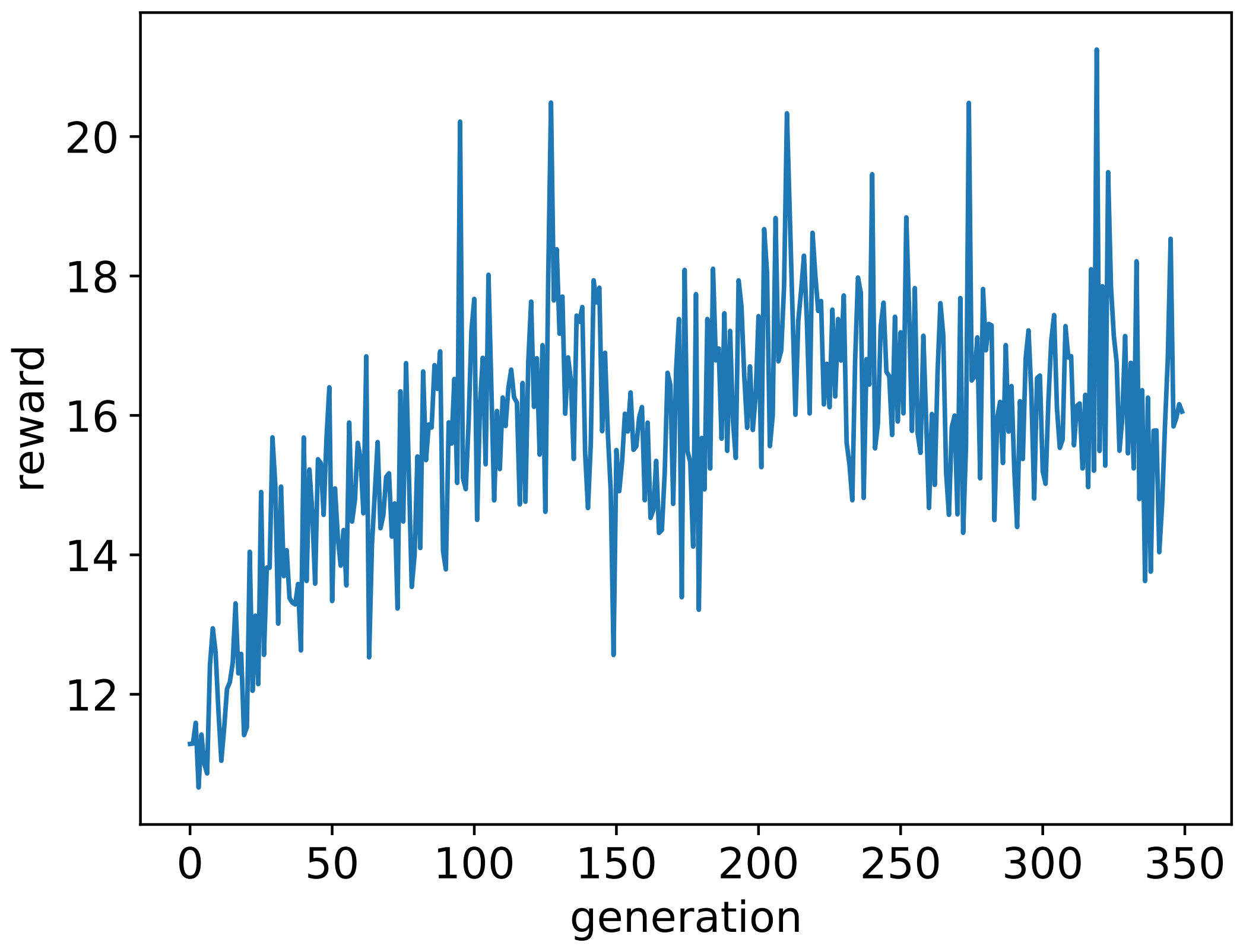


A genetic algorithm based approach can generate better solutions for more complicated tasks but requires much more training time and human parameter tweaking.

Experiment Results

Discrete world models tended to {out/underperform} continuous models as search depth increased. Discrete models had a {harder/easier} time representing more complex environments. Continuous models produced {better/worse} results when given equal train time due to having access to gradient based optimization.





Future Work

Future work involves further testing on complex environments to show more differences between models using continuous vs models using discrete representations and evaluating where improvements can be made. Developing a method to pin-point **where exactly** model degradation is occurring may also prove extremely useful when evaluating different methods of training.

References

Acknowledgments

This material is based upon work supported by the National Science Foundation under Grant No. 2050896.