Structure Learning in Motor Control: Investigations of LSTM Networks in Reinforcement Learning

Report for Research Internship II

M.Sc. Neuroscience, University of Freiburg

Hakan Yilmaz

Supervisor: Joschka Boedecker, Assistant Professor and head of Neurorobotics, Dep. of Computer Science, University of Freiburg

Project period: 21 May 2018 – 31 July 2018

Date of oral presentation: 30 April 2019, tbc

Submission date: 30 April 2019, tbc

Structure Learning in Motor Control: Investigations of LSTM Networks in Reinforcement Learning

Hakan Yilmaz[1,*]

[1]Master Student in M.Sc. Neuroscience, University of Freiburg, Germany
[*]hakan.yilmaz@students.uni-freiburg.de

ABSTRACT

While there is an abundance of experimental evidence for humans and animals learning and exploiting environmental structure to solve tasks both in a highly efficient and dexterous way, the underlying computational mechanisms are not yet well understood. Weinstein & Botvinick (2017) attempt to elucidate structure learning by using a model-based Deep Reinforcement Learning framework coupled with a cross-entropy planner in the domain of motor control. Using a Recurrent Neural Network as a model in this framework and through simulation experiments, the authors present a qualitative reproduction of a landmark study in motor structure learning by Braun et al. (2009). Despite of the remarkable model performance, in their study, not only the robustness of the results under variation of the training procedure or model and planner parameters remain unclear, but the authors also did not include a control group originally examined by Braun et al. (2009). In the present study, we attempt to address those shortcomings by first, reproducing and second, correspondingly augmenting the experiments performed by Weinstein & Botvinick (2017). While we could reproduce a similar qualitative difference between the groups investigated, we show differences between our and the authors' results suggesting that the model performances are sensitive to the training procedure and environmental simulation setup rather than variations of the model or planner parameters. Furthermore, our results suggest that the introduction of online learning is key towards a higher behavioral similarity between the model and human subjects.

Introduction

Both humans and animals possess the remarkable ability to quickly adapt to novel domains and to solve novel tasks both in a highly efficient and dexterous way. One requirement for this ability is the (unconscious or conscious) recognition and memorization of task structure that guide learning and allow agents to efficiently solve tasks with similar structure, a process called structure learning (Braun et al., 2010). While structure learning can be observed across several domains ranging from high-level cognition in humans (Tenenbaum et al., 2011) to motor control (Braun et al., 2010) through to the adaptation of visuomotor reflexes (Kobak & Mehring, 2012) all of where structure is exploited while perceiving the environment as well as acting on it (Gershman & Niv, 2010), the underlying computational mechanisms are not yet well understood. A better understanding not only promises guidance regarding the neural implementation of structure learning, (Tervo et al., 2016; Lansdell & Kording, 2019) but also has a high potential to bring the field of Artificial Intelligence closer to more autonomous agents.

One popular investigation approach comes from a Bayesian perspective where structure learning consists of the acquisition of a generative model (or several competing generative models) modeling a relevant task domain as well as the refinement of its parameters (Tenenbaum et al., 2011; Braun et al., 2009; Genewein et al., 2015). Another approach comes from a deep learning perspective where Recurrent Neural Networks (with the inherent ability to preserve temporal dynamic structure) in the frame of model-free Reinforcement Learning (RL) have successfully been used to exploit task structure and to consequently increase agent performance (Wang et al., 2016).

In this study however, we focus on a recent idea by Weinstein & Botvinick (2017) who proposed a model-based Reinforcement Learning framework that builds upon both of the aforementioned perspectives. In this framework (Fig. 1C), a generative model $\hat{M}$ is queried by a planner P providing predicted reward-observation-trajectories $(r_{1..T+1}, o_{1..T+1})$ from a current observation o and a sequence of actions $a_{0..T}$, where T is the trajectory length. The key contribution of this framework is the model-internal memory α, realized by a LSTM network (Hochreiter & Schmidhuber, 1997), which allows the system to perform system identification from observations, i.e. to identify the actual task domain M from an arbitrary large number n of task domains M_n characterized by varying underlying task structure.

Another key contribution by Weinstein & Botvinick (2017) is the application of this framework to structure learning in motor control qualitatively reproducing the results of a landmark experiment in motor structure learning by Braun et al. (2009). In this experiment, human subjects were trained performing a goal-reaching task in environments of varying underlying visuomotor transformation: While the *Rot-Learner* group only experienced rotations of the visual field, the *Random* group was exposed to arbitrary linear visual transformations. In the subsequent testing phase, in which all subjects (including those from a *Control* group) were subject to rotations of ±60°, the *Rot-Learner* group showed faster adaptation compared to the other groups (see Fig. 1A). Braun et al. (2009) concluded that subjects in the *Rot-Learner* group must have learned the underlying structure of the test environment, namely the presence of rotations, more efficient than subjects experiencing either more complex structure (i.e. *Random* group) or differing structure (i.e. *Control* group).

Weinstein & Botvinick (2017) presented concordant results after training models in two of the three aforementioned task environments using their framework (Fig. 1C), *Rot* as the rotation-only environment and *Rot+* as the random-transformation environment, and subsequently testing the models similar to Weinstein & Botvinick (2017) albeit various simplifications (see results in Fig. 1B).

However, although the authors clearly state a minimal modeling and simulation approach focusing on the structure learning effect between the investigated groups, the details of the deep learning model, the simulation environment and the cross-entropy planner remain obscure. Consequently, accompanied by the question of reproducibility, questions about the model/planner behavior under varying parametrization arise naturally. Furthermore, the authors omit to discuss the worsening model performance in the late trials of their experiments (Fig. 1B), calling for investigations beyond the five trials. Finally, for reasons not further specified, the authors did not include the *Control* group in their experiments despite the interesting behavior of starting with a poor testing performance as

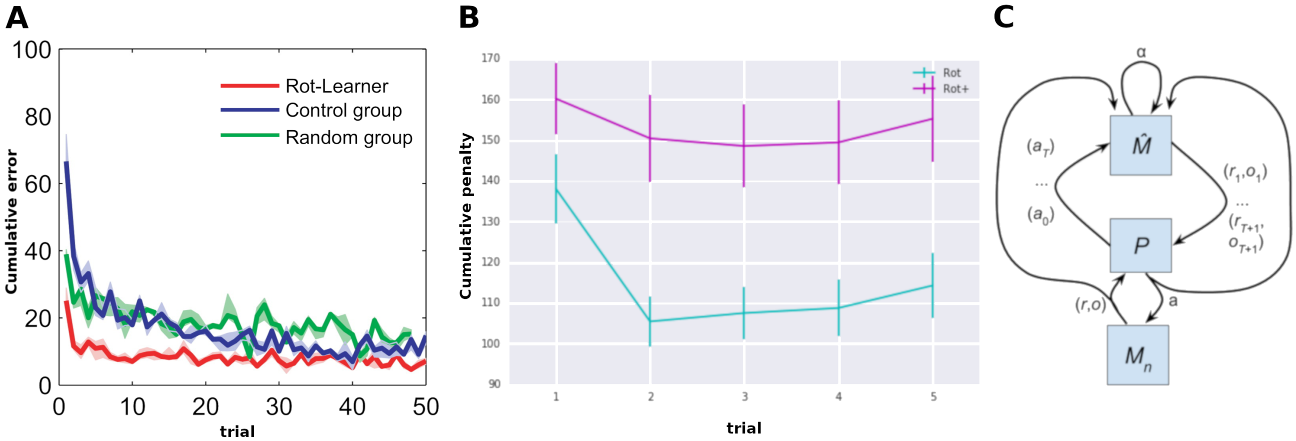

Figure 1. **A**. Test results in 60° rotation environment from human behavioral experiments by Braun et al. (2009); figure adapted. **B**. Test results in 60° rotation environment from simulation experiments by Weinstein & Botvinick (2017); figure adapted. **C**. Model-based reinforcement learning framework with the ability to perform structure learning proposed by Weinstein & Botvinick (2017).

opposed to the other groups and eventually performing at least comparably in the original experiments (Fig. 1A).

In this study, we address the aforementioned shortcomings. We first show the results from our reproduction attempts along with an analysis of the planner's robustness. Following this, we conduct augmented experiments that include the *Original* control group, a higher number of trials during experiments and variations of the capacity of the LSTM network, the most critical part of the framework as it effectively forms its memory. We conclude with an outlook on extensions of this model and further potential applications thereof.

Methods

For our experiments, we implemented a simplified arm model using the physical simulation engine MuJoCo Pro 1.5 (Todorov et al., 2012). We assumed upper arm and forearm lengths of 30 cm each with minimal natural limitations, e.g, an over-extension of the elbow was not possible (see arm model in Fig. 2). We further assumed maximum shoulder and elbow joint torques of ± 10 N m.

Similar to Weinstein & Botvinick (2017), we first trained deep neural networks (one LSTM layer followed by two fully connected layers with 100 units each) on 2000 random walk trajectories of step length 42 (corresponds to 3 s simulation time) in three different environments: In the first environment called *Rot*, the visual feedback of the hand position was rotated by a rotation angle uniformly drawn between -90° and 90°. In the second *Rot+* environment, instead of a pure rotation, the hand position was subject to arbitrary linear transformations, i.e. combinations of as shearings, scalings and rotations (see Braun et al. (2009), Supplemental Material). In the third *Original* control environment, feedback was veridical.

Training was designed to predict x-y-coordinates as well as x-y-velocity components from previously visited/taken coordinates, velocities and actions (i.e. accelerations). Furthermore, training trajectories were not always complete, but interrupted by environmental resets, i.e. resets of the hand position to $x = y = 0$ cm. During a trajectory, a reset (denoting the end of a trial) happened stochastically with a probability of 1/30 in each step, and was indicated by a high reset bit included in the training procedure. This way, according to the authors who shared using this procedure by e-mail, the LSTM network is expected to incorporate environment resets into its

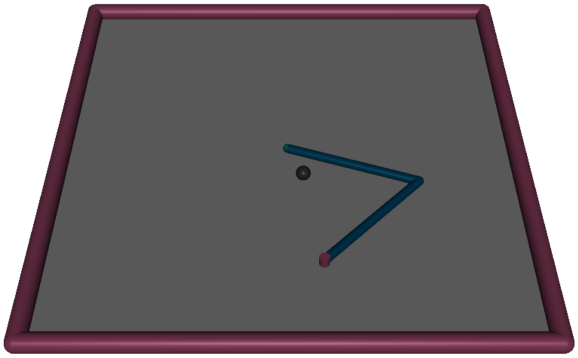

Figure 2. Simulation environment in MuJoCo. Cylinder represents fixed shoulder position, gray sphere represents goal region.

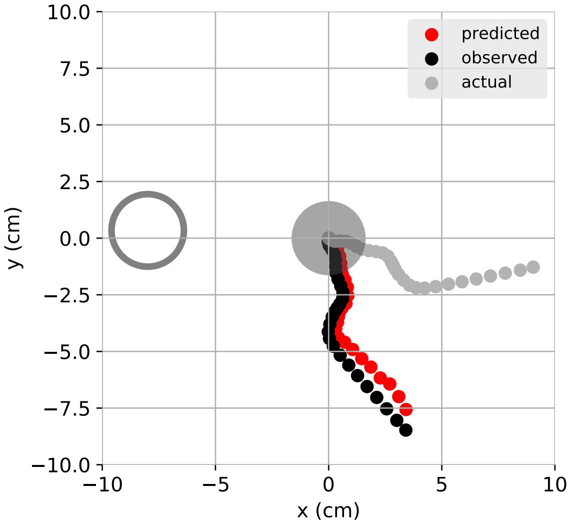

Figure 3. Example random walk trajectory in the test environment for a model trained in the *Original* environment. Gray ring represents goal region.

hidden states.

In addition to experiments with 100 LSTM units originally used by Weinstein & Botvinick (2017) and in order to learn about model behavior under varying memory capacity, we trained and tested models with 50, 150 and 200 units as well. We implemented our network in TensorFlow (Abadi et al., 2016) using the Adam optimizer (Kingma & Ba, 2014) in 4000 epochs.

In order to eventually test the trained models in the 60° rotation environment similar to Weinstein & Botvinick (2017), we implemented the general-purpose cross-entropy planner proposed by Weinstein & Littman (2013). In each time step, this planner draws random action pairs from a proposal distribution, performs roll-outs of these actions (i.e. queries reward and trajectory predictions from a generative model, where trajectory length is called horizon and the batch size corresponds to how many trajectories are evaluated), refines the parameters of the proposal distribution based on trajectories with high reward expectation and repeats this procedure for a predefined number of generations. In order to analyze the robustness of this planner and to find a robust parameter set for our model experiments, we first tested the planner in ground-truth with varying parameters (distribution type, batch size, horizon length, number of generations). We emulated model uncertainty by imposing relatively high Gaussian motor noise with standard deviation 5 N m on the actions used for ground-truth roll-outs. As a result, we selected a planner with action proposal according to individual Gaussians, batch size 100, horizon length 20 and 10 generations per planning step for our model experiments. As opposed to Weinstein & Botvinick (2017), we did not use warm-starting since we could observe trajectory variance collapsing even in the early steps of a trial.

In the testing phase, we coupled the trained models with the planner and performed 100 experiments for each model in the test environment, where visual feedback was rotated by 60° (see example trajectory for a *Original* model in Fig. 3). We additionally increased trial length from 5 to 7 compared to Weinstein & Botvinick (2017), with each trial either terminating after a maximum of 2 s of simulation time or terminating early if the observed hand position was within a radius of 1.6 cm from the goal center for more than 500 ms. The main measure for performance that Weinstein & Botvinick (2017) used in their experiments, was cumulative penalty, i.e. the cumulative observation-to-goal distance per simulation step in one trial. We additionally measured cumulative error, i.e. the cumulative distance between observations and the straight line between start position and goal center, which was originally used by Braun et al. (2009) and which is similar, but not equivalent to cumulative penalty.

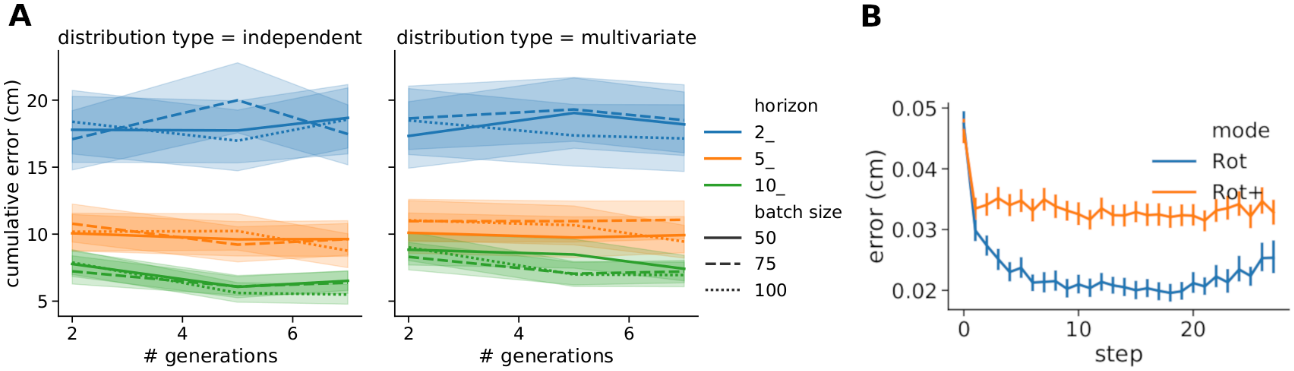

Figure 4. **A**. Analysis of planner robustness and performance. Planning was performed using ground-truth predictions subject to motor noise. **B**. Model prediction error per time step for random walks after training (without planner); 95 % CI presented.

Results

We first investigated planner performance and robustness in ground-truth with relatively high planning noise of 5 N m in order to emulate prediction errors by the generative model queried by the planner. For each parameter combination, we performed 100 experiments and planning was successful for all combinations. In that sense, we found high planner robustness to parameter variation, where the horizon length, i.e. how many open-loop time steps are taken into account at each time step, has the highest impact on planning performance (Fig. 4A). Similar to Braun et al. (2009), we used cumulative error, i.e. cumulative deviation from observed positions to the straight line from start to goal as a measure for performance. For all subsequent model experiments, we used this planner with independent Gaussians as proposal type and a batch size of 100. We further bolster anticipated model performance by using 10 generations per step and horizon length 20.

Following this, we started our attempts to reproduce the results by Weinstein & Botvinick (2017). We first trained two models with 100 LSTM units on random walk trajectories in the *Rot* and *Rot+* environments. Training was successful, leading to an average prediction error of 0.027 cm for the *Rot* model and 0.048 cm for the *Rot+* model for all training trajectories. Although we could not support the low prediction errors for training data that the authors obtained after training, we could qualitatively reproduce model behavior on random walks without planning in the test environment, i.e. 60° rotation (Fig. 4B): Prediction error per time step sharply decreases early in the trajectory, providing evidence for both models successfully performing system identification with increasing performance as more observations in the test environment are added. The most important observation however, is the performance difference between *Rot* and *Rot+*. Despite training on the same amount of data and even on (stochastically) the same number of pure rotations around ± 60°, the *Rot* model consistently outperforms *Rot+* by approx. 0.01 cm per time step. Also, similar to Weinstein & Botvinick (2017), we could observe an error increase for the last steps of *Rot* trajectories. Yet, in contrast to the authors' results, the prediction errors are significantly higher. A possible explanation for the quantitative difference between the authors' and our results could lie in different physical simulation setups (e.g., a more constrained workspace is expected to result in smaller prediction errors) or training setups (potentially arising from differences in availability of hardware resources) none of which is further specified by the authors.

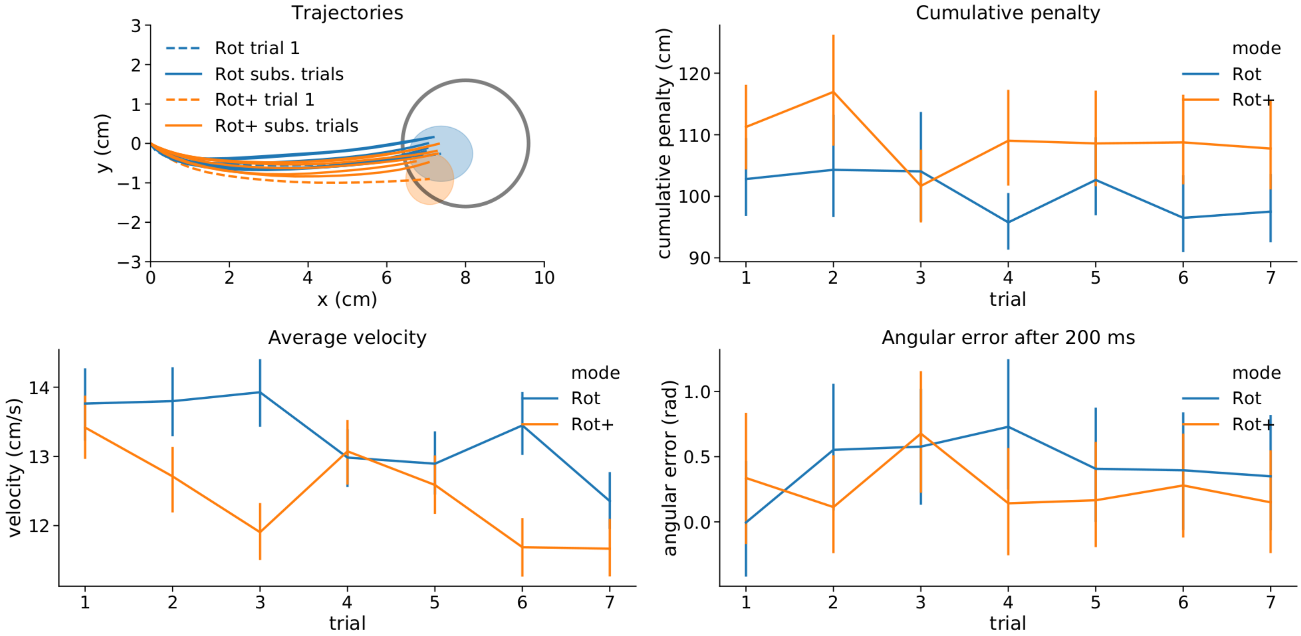

Figure 5. Results for experiments with *Rot* and *Rot+* models in combination with the cross-entropy planner, compare with Weinstein & Botvinick (2017); 95 % CI presented.

As a next step, we coupled the models with the planner and performed 100 experiments in the test environment for each model. Similar to Weinstein & Botvinick (2017), we measured not only cumulative penalty over trial reflecting model performance, but also average hand velocity per trial and angular error after 200 ms in each trial. Our results significantly deviate from the original results by the authors (see our results in Fig. 5 and compare with Fig. 4 and 5 in Weinstein & Botvinick (2017)): First, while the performance difference between the models seems consistent with the results in Fig. 4B, it is less significant, exemplified by the cumulative penalty in the third trial. While the average difference here is approx. 10 cm, the difference in Weinstein & Botvinick (2017) of approx. 40 units is more pronounced. Second, the difference in angular error for both models is non-significant. The third and most striking deviation is the model performance in the course of the experiment. While the authors' results suggest that performance not only increases within one trial, but also from one trial to another (marking the reproduction success w.r.t. the original experiments by Braun et al. (2009)), we could not observe this effect in our experiments. Instead, performance does not significantly change. This is reflected in highly clustered albeit highly precise trajectories.

Following this, we asked the question of how model behavior changes under variation of memory capacity, i.e. under a varying number of LSTM units per model. For the corresponding experiments, we included models trained in the control environment called *Original* (equivalent to *Control*), where visual feedback was veridical. The results for prediction errors for random walks without planning in the test environment are consistent with the intuition that for increasing model capacity prediction error generally decreases (Fig. 6). An interesting observation in our experiment however, is the ample failure of the *Original* model to correctly predict observations. An intuitive explanation for that failure is the absence of online learning, marking a significant difference between how the herein used models and humans learn tasks: While for the model a clear distinction between training and testing is possible by freezing its weights, humans evidently show the ability to extract and learn task structure in a highly efficient manner allowing the *Control* group (equivalent to our *Original* model) in Braun et al. (2009) to compete with the other groups during testing.

We subsequently coupled the trained models with the planner and measured cumulative penalty, angular error after 200 ms as well as trajectories similar to Fig. 5. As expected, because of too high prediction errors as seen in Fig. 6, planning was not successful with the *Original* model, reflected by consistently high cumulative penalty (Fig. 7A). In contrast, planning was successful with *Rot* and *Rot*+ models with low (50 units) as well as high (200 units) capacity. While we could observe a high similarity in performance between all models, a performance increase with increasing capacity becomes apparent which is especially noticeable in Fig. 7C, where trajectory precision increases and variability decreases. An exception however is the *Rot*+ model with 200 LSTM units, for which cumulative penalty increases sharply. The angular error after 200 ms does not reveal differences between models. In fact, the comparable angular errors with the *Original* environment indicate that measuring angular error after 200 ms (i.e. 3 simulation steps) might be too early supporting the hypothesis that simulation or training differences between our approach and that of Weinstein & Botvinick (2017) lead to deviations in observable effects and effect sizes. For interpreting the results obtained by the authors it should be taken into account

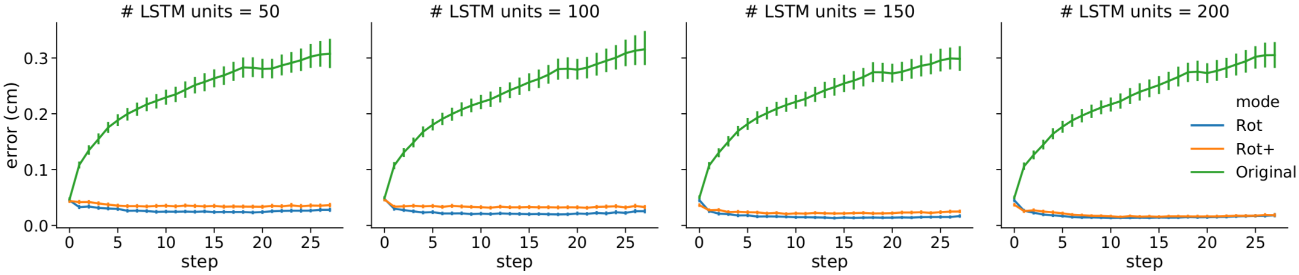

Figure 6. Model prediction error per time step for random walks after training (without planner); 95 % CI presented.

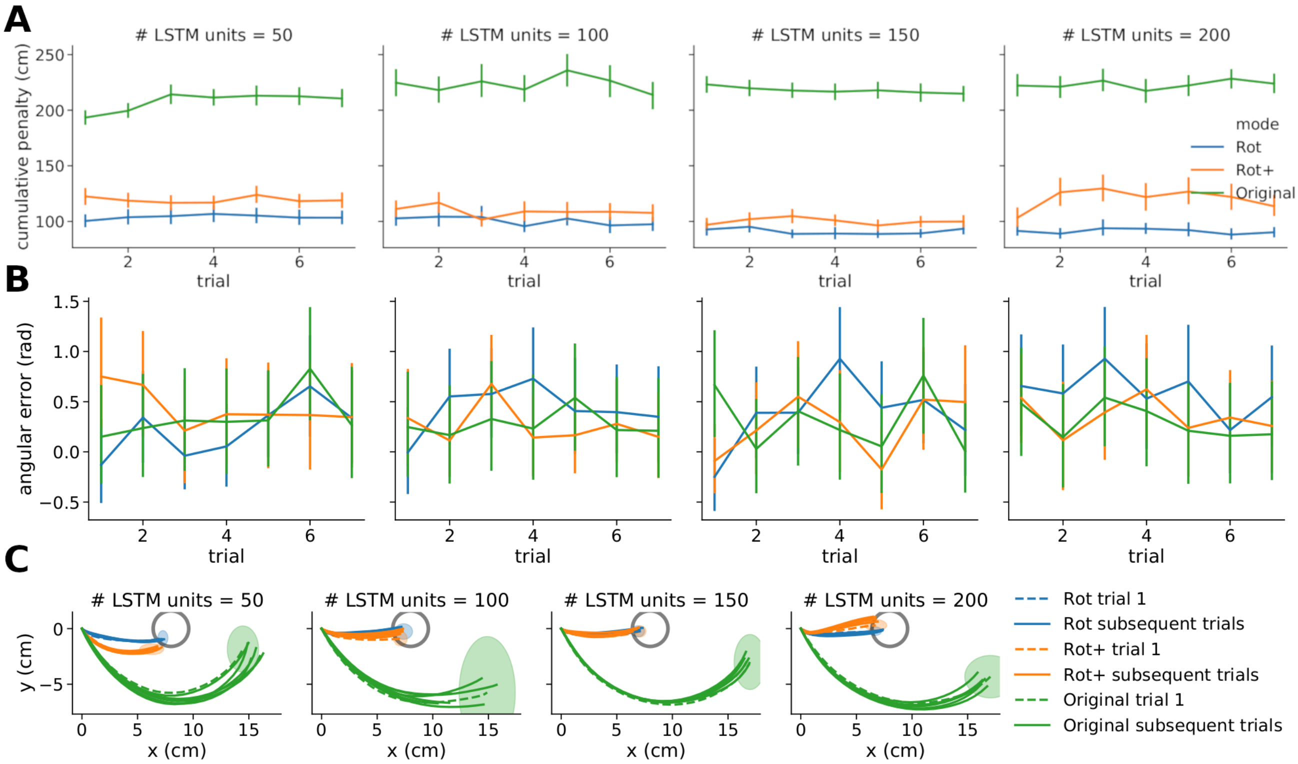

Figure 7. **A**. Cumulative penalty over trials for all model capacities and environmental modes during testing. **B**. Angular error after 200 ms. **C**. Average trajectories. 95 % CI presented in A and B.

that cumulative penalty is similar, but not equivalent to the cumulative error (i.e. cumulative distance between individual observations and the straight line between start and goal) used by Braun et al. (2009). We measured cumulative error (see attached Fig. S.1), where effect sizes between groups appear even smaller.

Discussion

In this study, we investigated a model-based reinforcement learning framework proposed by Weinstein & Botvinick (2017) effectively combining two existing computational approaches to structure learning, a Bayesian perspective and a model-free deep reinforcement learning perspective. One immediate strength of the authors' study is the apparent reproduction of results from a landmark study on motor structure learning by Braun et al. (2009), in which training of models in environments with different structure (pure visual rotation *Rot* as opposed to arbitrary linear transformation *Rot+*) leads to increased performance during testing when task structure in training and testing (60° rotation) is similar.

In order to answer questions on this framework about reproducibility, planner/model robustness and model behavior under variation of memory capacity, we could first show a high planner robustness extracting parameters for which planning with LSTM networks after training on random walk data showed to be successful. Second, we attempted a reproduction of the results obtained from Weinstein & Botvinick (2017), presenting significant deviations between our results and those of the authors such as seemingly constant model performance over trials and only small effect sizes between the models. An explanation for those deviations potentially lies in the physical simulation setup, where variations of the arm model including the range of joint acceleration and movement restrictions might have a big impact on training and testing, as well as in the training procedure: For example, in our experiments, we did not only predict x-y-coordinates, but also the x-y-components of the transformed hand velocity which Weinstein & Botvinick (2017) might have not explicitly predicted (in our experiments, planning without explicit prediction of velocity was not possible as individual prediction errors were too high). The observation of highly clustered and precise trajectories supports this hypothesis since an increased number of training data and an increased number of information coming in during testing is likely to promote system identification - in our case potentially up to the point where the ability to perform system identification leads to vanishing differences between *Rot* and *Rot*+ and across trials. In that sense, increasing observation prediction performance by including velocity prediction would correspond to the authors' observation that performance differences between *Rot* and *Rot*+ essentially vanishes if the data corpus is expanded by a factor of five. Another interesting observation was the sharp decrease in performance of the *Rot*+ model with 200 LSTM units: A possible explanation for this admittedly somewhat weak effect (as it cannot be observed for 150 LSTM units) is that a higher memory capacity trades off precision and data requirement. It is possible that a certain capacity, e.g. 200 LSTM units, the data requirement dominates this trade-off. Further investigations for even higher capacities are likely to elucidate this issue.

In conclusion, it is likely that the wide differences between the results from Weinstein & Botvinick (2017) and those from our reproduction and augmentation attempts stem from a difference in experimental setup rather than from a inherent non-reproducibility or lack of model robustness. However, we believe that an extension of this model, the ability to perform online-structure-learning, has the potential of increasing generalisability not only to arbitrary setups such as that used in our experiments, but also to the *Control* group which Weinstein & Botvinick (2017) did not investigate. Another interesting research direction lies in the interpretability of a generative model instantiated by a LSTM network which is amenable to analyses similar to those performed by Karpathy et al. (2015), potentially providing experimental and theoretical impulses towards bridging the gap between Bayesian methods and deep learning.

References

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mane, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viegas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., & Zheng, X. (2016). TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems. *arXiv*.

Braun, D. A., Aertsen, A., Wolpert, D. M., & Mehring, C. (2009). Motor Task Variation Induces Structural Learning. *Current Biology*, *19*(4), 352–357.

Braun, D. A., Mehring, C., & Wolpert, D. M. (2010). Structure learning in action. *Behavioural Brain Research*, *206*(2), 157–165.

Genewein, T., Hez, E., Razzaghpanah, Z., & Braun, D. A. (2015). Structure Learning in Bayesian Sensorimotor Integration. *PLOS Computational Biology*, *11*(8), 1–27.

Gershman, S. J., & Niv, Y. (2010). Learning latent structure: Carving nature at its joints. *Current Opinion in Neurobiology*, *20*(2), 251–256.

Hochreiter, S., & Schmidhuber, J. (1997). Long Short-Term Memory. *Neural Computation*, *9*(8), 1735–1780.

Karpathy, A., Johnson, J., & Fei-Fei, L. (2015). Visualizing and Understanding Recurrent Networks. *arXiv*.

Kingma, D. P., & Ba, J. (2014). Adam: A Method for Stochastic Optimization. *arXiv*.

Kobak, D., & Mehring, C. (2012). Adaptation Paths to Novel Motor Tasks Are Shaped by Prior Structure Learning. *Journal of Neuroscience*, *32*(29), 9898–9908.

Lansdell, B. J., & Kording, K. P. (2019). Towards learning-to-learn. *arXiv*.

Tenenbaum, J. B., Kemp, C., Griffiths, T. L., & Goodman, N. D. (2011). How to grow a mind: Statistics, structure, and abstraction. *Science*, *331*(6022), 1279–1285.

Tervo, D. G. R., Tenenbaum, J. B., & Gershman, S. J. (2016). Toward the neural implementation of structure learning. *Current Opinion in Neurobiology*, *37*, 99–105.

Todorov, E., Erez, T., & Tassa, Y. (2012). MuJoCo: A physics engine for model-based control. In *IEEE International Conference on Intelligent Robots and Systems*, (pp. 5026–5033). IEEE.

Wang, J. X., Kurth-Nelson, Z., Tirumala, D., Soyer, H., Leibo, J. Z., Munos, R., Blundell, C., Kumaran, D., & Botvinick, M. (2016). Learning to reinforcement learn. *arXiv*.

Weinstein, A., & Botvinick, M. M. (2017). Structure Learning in Motor Control:A Deep Reinforcement Learning Model. *arXiv*.

Weinstein, A., & Littman, M. (2013). Open-Loop Planning in Large-Scale Stochastic Domains. In *AAAI Conference on Artificial Intelligence*, (pp. 1436–1442).

Additional information

This paper is part of the Research Internship II as a component of the M.Sc. Neuroscience program at the University of Freiburg. Code and the online version of this paper are available at `https://github.com/HakYi/Structure-Learning-DRL-Model`.

Attachments

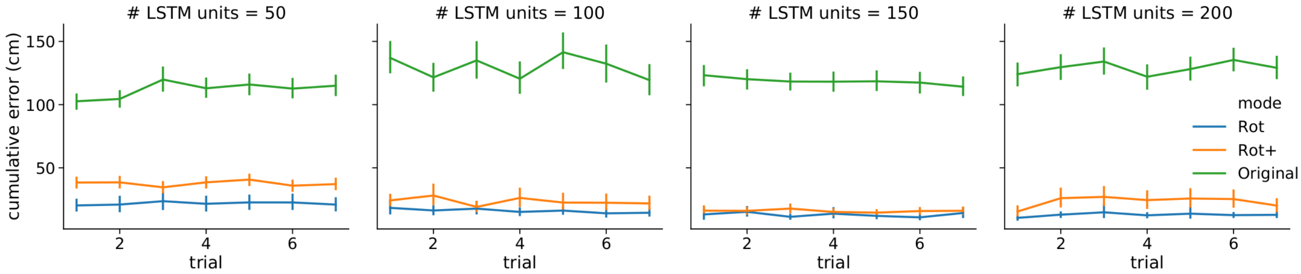

Figure S.1. Cumulative penalty over trials for all model capacities and environmental modes during testing; 95 % CI presented.