Likelihood Approximation Networks enable fast estimation of generalized sequential sampling models as the choice rule in RL

Krishn Bera
Dept. of Cognitive, Linguistic & Psychological Sciences
Carney Institute for Brain Science
Brown University
krishn_bera@brown.edu

Alexander Fengler
Dept. of Cognitive, Linguistic & Psychological Sciences
Carney Institute for Brain Science
Brown University
alexander_fengler@brown.edu

Michael J. Frank
Dept. of Cognitive, Linguistic & Psychological Sciences
Carney Institute for Brain Science
Brown University
michael_frank@brown.edu

Abstract

Sequential sampling models (SSM) are a powerful class of models used to summarize cognitive process dynamics underlying decision-making in various task settings. In reinforcement learning (RL), while researchers typically assume a simple softmax choice rule, more recently, studies have used the drift diffusion model (DDM), a popular SSM to jointly model choice and response time distributions during learning (Pedersen, Frank, & Biele, 2017; Pedersen & Frank, 2020; Fontanesi, Gluth, Spektor, & Rieskamp, 2019). Such an approach allows researchers to study not only the across-trial dynamics of learning but the within-trial dynamics of choice processes, using a single model.

However, a practical problem in parameter estimation is the lack of closed-form likelihoods for a large class of models. Such intractable likelihoods render typical Bayesian inference methods infeasible. Alternative likelihood-free inference methods need to be invoked, which often tend to incur enormous computational costs, thereby limiting their application. To enable Bayesian estimation for a broad class of RL-SSM models, we leverage the recently developed Likelihood Approximation Networks (LAN) (Fengler, Govindarajan, Chen, & Frank, 2021). The LAN approach involves training neural networks that learn approximate likelihoods for arbitrary generative models, allowing fast posterior sampling with only a one-off cost for model simulations that are amortized for future inference. Once amortized, the approximate likelihoods can be used for tractable inference via MCMC across arbitrary experiment designs, while allowing a much larger class of SSMs to serve as the behavior generating mechanisms of reinforcement learning agents.

Using synthetic datasets, we show here, a proof of concept that this method can be utilized to estimate the true posterior parameter distributions for the RL-DDM. Furthermore, we show accurate parameter recovery in hierarchical settings. We conclude by proposing a LAN-based reinforcement learning extension to the widely used HDDM Python toolbox (Wiecki, Sofer, & Frank, 2013), which would allow us to leverage LANs for arbitrary RL-SSM models.

Keywords: Sequential Sampling Models, Reinforcement Learning, Parameter Estimation, Approximate Bayesian Computation

Acknowledgements

This work was funded by NIMH grant P50 MH 119467-01.

1 Introduction

Hybrid computational models, combining the conceptual ideas of sequential sampling of evidence with reinforcement learning models have greatly expanded the cognitive modeller's toolbox in recent years (Pedersen et al., 2017; Fontanesi et al., 2019). The main idea behind these models is to allow a reinforcement learning (RL) process to drive the trial-by-trial parameters of a sequential sampling model (SSM) such as the basic drift diffusion model (DDM) (Ratcliff & McKoon, 2008), to jointly capture reaction time and choice behavior in complex tasks which involve learning from feedback (see Figure 1). As a result, RL-SSM is a much more powerful and expressive class of models. It naturally lends itself for use in computational modeling of numerous cognitive tasks where the 'learning process' informs the 'decision-making process'.

The combination of these two classes of models however inherits a shortcoming which plagues the world of sequential sampling models to begin with. Great interest in model variants, such as models which move beyond 2-choice scenarios, models which exchange the noise distribution (Wieschen, Voss, & Radev, 2020) in the diffusion, models which include attractor dynamics (Usher & McClelland, 2001) as well as models which deal with non-constant evidence criteria (Cisek, Puskas, & El-Murr, 2009), is contrasted with empirical data analysis being mostly limited to basic versions of the drift diffusion model, which uses evidence criteria that are constant over time as well as a linear accumulation process perturbed by Gaussian noise. This state of affairs falls out of the simple analytic convenience provided by the basic DDM, for which fast-to-compute likelihood functions (Navarro & Fuss, 2009) exist to make inference tractable, while no such closed-form likelihood functions exist for most variations of interest.

Hence, while recent work has enabled the development of easy to use software to combine the DDM and RL (Pedersen & Frank, 2020; Fontanesi, 2021), the natural extension to a combination of RL with a larger class of SSMs has been hampered by the lack of easy to compute likelihoods. The main bottleneck is formed by the need for expensive model simulations during inference, making statistical inference for such models intractable for anyone but highly computationally sophisticated experts.

Recent advances in the field of likelihood-free inference (Papamakarios & Murray, 2016; Gutmann, Dutta, Kaski, & Corander, 2018; Papamakarios, Nalisnick, Rezende, Mohamed, & Lakshminarayanan, 2019; Papamakarios, Sterratt, & Murray, 2019; Lueckmann, Bassetto, Karaletsos, & Macke, 2019; Radev, Mertens, Voss, & Köthe, 2020; Fengler et al., 2021), provide a suite of new tools that utilize the power of deep learning to improve the computational efficiency of statistical inference in models which are accessible only via simulations (but lack a analytical likelihood function).

We leverage here one such tool, likelihood approximation networks (LANs) (Fengler et al., 2021), to bridge the gap towards the application of a wider class of SSMs conjointly with RL. In this preliminary work, we provide a proof of concept that LANs can be used fruitfully for this type of modelling.

In the 'Methods' section, we provide a brief description of the specific LAN that we employed and how it was trained. The next subsection outlines the test bed which we use to generate the synthetic dataset and run parameter recovery. The 'Results' section shows proof-of-concept parameter recovery and the results of our approach when applied to hierarchical settings. We conclude by highlighting the significance/benefits of the proposed method and outlining the future directions.

2 Method

2.1 Likelihood Approximation Network

We trained a multilayer perceptron with three layers to approximate the likelihoods, as per the procedures outlined in the LAN reference paper (Fengler et al., 2021). The resulting network takes the data and model parameters (features during training) as input and outputs trial-wise (approximate) log-likelihood values (labels, based on kernel density estimates of empirical likelihood functions). The specific network used in this article was trained on 3×10^5 parameter combinations (with 2×10^5 simulations of each run) of the drift-diffusion model using Pytorch (Paszke et al., 2019).

2.2 Test Bed

We test our method on synthetic datasets of the multi-armed bandit task. The experiment is a two-armed bandit task with binary outcomes. The model employed a simple delta learning rule (Rescorla, 1972) to update the action values

$$q_{action}(t+1) = q_{action}(t) + \alpha * [r(t) - q_{action}(t)],$$

where $q_{action}(t)$ denotes expected reward (Q-value) for the chosen action at time t, $r(t)$ denotes reward obtained at time t and α denotes the learning rate. As an initial benchmark to compare with analytical solution, we began with using a LAN for the vanilla two-choice drift-diffusion model with three parameters - boundary seperation (a), non-decision time

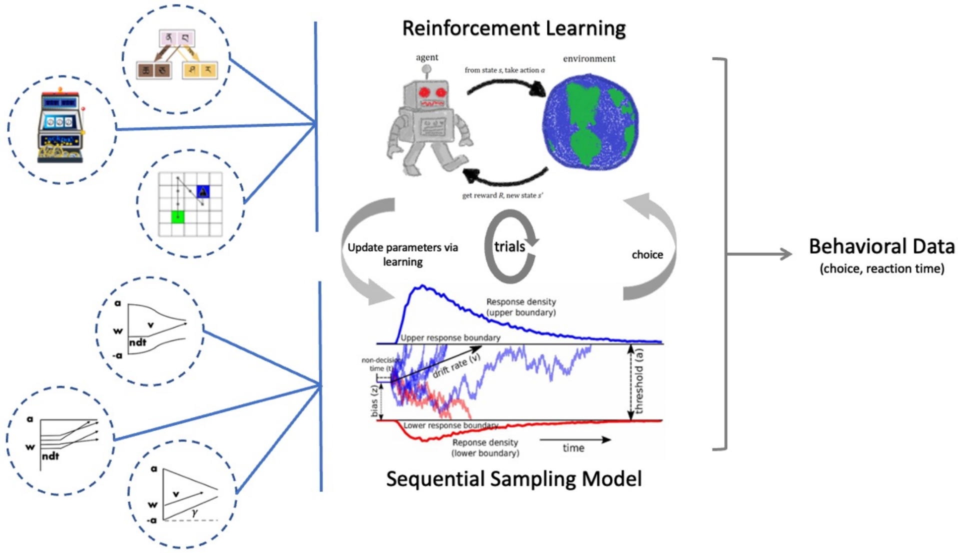

Figure 1: RLSSM - combining reinforcement learning and sequential sampling models.

(t) and drift rate (v). The trial-by-trial drift rate depends on the expected reward value learned by the RL rule. The drift rate is therefore a function of Q-value updates, and is computed by the following linking function

$$v = (q_{action1} - q_{action2}) * s,$$

where s is a scaling factor of the difference in Q-values. In other words, the scaler s is the drift rate when the difference between the Q-values of both the actions is exactly one.

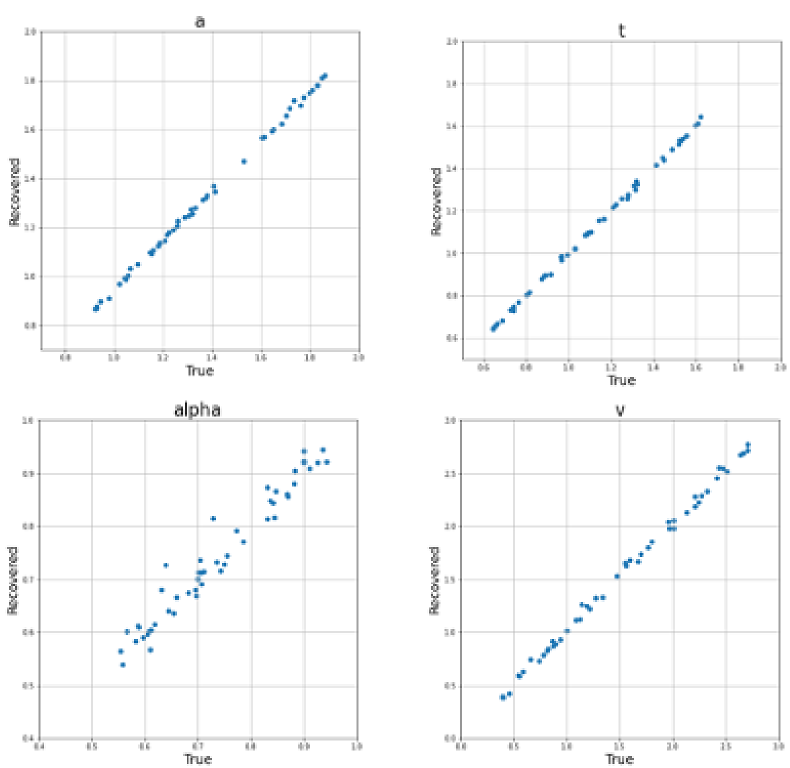

Figure 2: Parameter recovery on synthetic datasets. The true group mean parameter values (used to generate the datasets) are plotted on the x-axis. The recovered group mean parameter values are plotted on the y-axis. a is boundary separation, t is non-decision time, $alpha$ is the learning rate and v is the scaling factor for drift rate.

3 Results

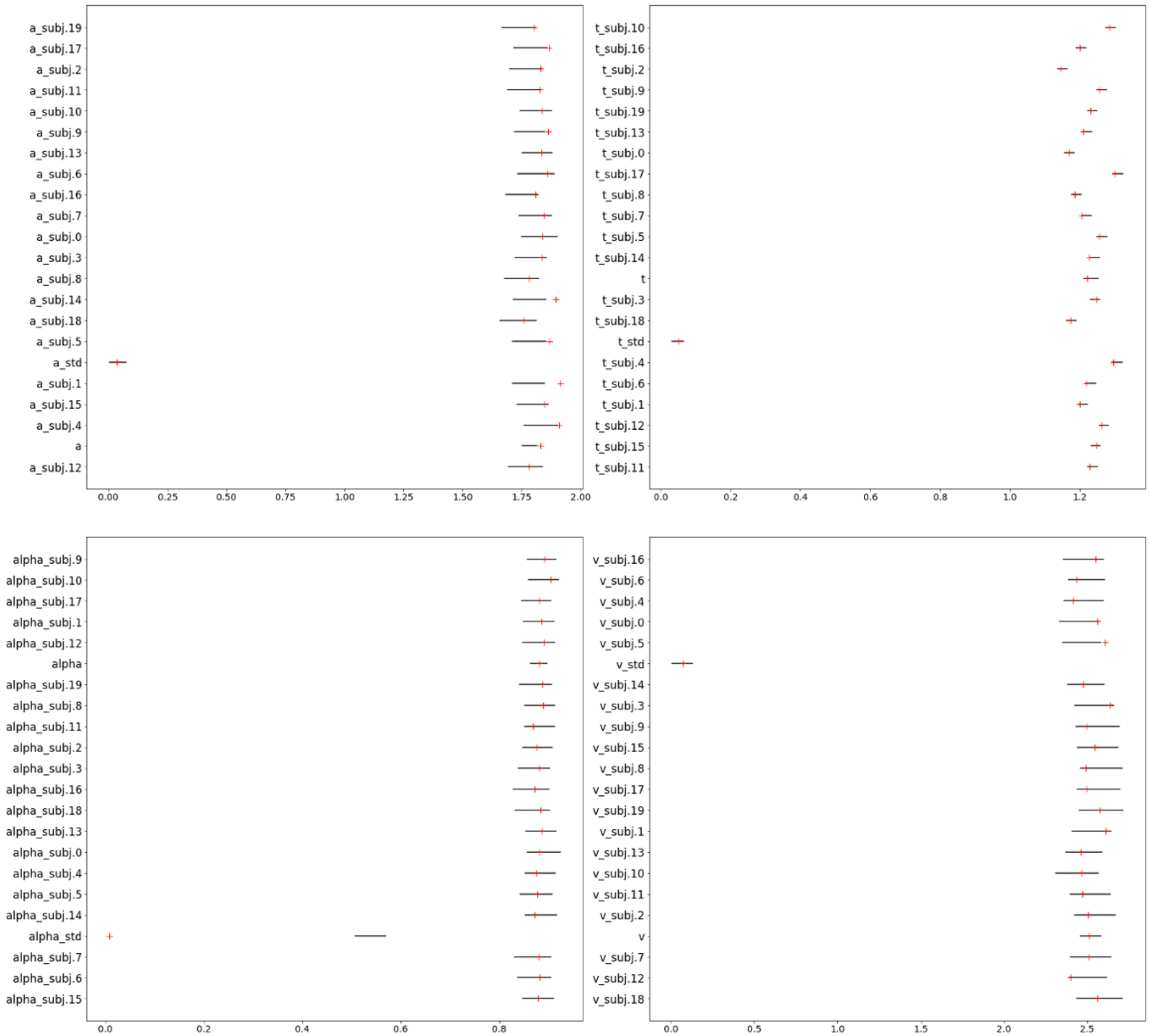

Figure 3: Hierarchical parameter recovery. Posterior distributions (denoted by caterpillar plots) of recovered parameters of a dataset with 20 subjects. The black line corresponds to 90% highest density interval. The ground truth parameter values are denoted by red crosses. Group mean and standard deviation are also plotted (for ex. see a and a_std).

3.1 Parameter Recovery

For parameter recovery experiments, 50 synthetic datasets were generated using the RL-DDM model. Each dataset contained 20 subjects with 500 simulated trials per subject. The mean and standard deviation of group parameters were sampled from a uniform distribution over a reasonable range of parameters. The subject means were sampled from a truncated normal distribution, which was parameterized by group mean and standard deviation. The probabilities of reward for choosing the action corresponding to the upper and lower boundary were 0.8 and 0.2, respectively. To generate samples from the respective posterior distributions, we employed MCMC, specifically coordinate-wise slice sampling (Neal, 2003) as implemented in the HDDM toolbox (Wiecki et al., 2013). Figure 2 shows parameter recovery on the generated datasets.

3.2 Application to Hierarchical Settings

Figure 3 shows hierarchical parameter recovery on a sample dataset. We show that the inference method can be generalized to simultaneously estimate reinforcement learning parameters and decision parameters within a fully hierarchical Bayesian estimation framework. The LAN-based inference was able to accurately recover individual and group level parameters.

3.3 Posterior Predictive Checks

An important step in computational modeling is validating the model at hand. We check for model validity using posterior predictive checks, which involves simulating data using estimated parameters and comparing observed and simulated results. The simulated dataset was obtained by repeating the simulation process 500 times for each subject in a sample dataset. To evaluate the choice proportion for best option across learning for observed and simulated data, we bin the trials and plot 90% highest density intervals of the mean responses. Figure 4 shows mean response rate and reaction time densities in simulated and observed datasets.

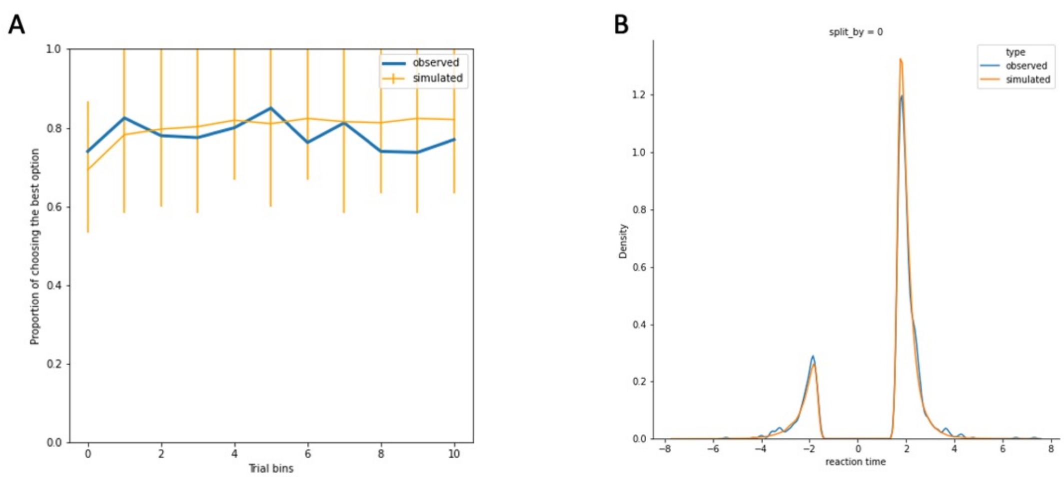

Figure 4: Posterior predictive checks. (A) Rate of choosing the best option across learning. Uncertainty in the generated data is captured by the 90% highest density interval of the means across simulated datasets. (B) Density plots of observed and predicted reaction time across conditions. RTs for lower boundary choices (i.e. worst option choices) are set to be negative (0-RT) to be able to separate upper and lower bound responses

4 Conclusion and Future Directions

The present work serves as a basic proof of concept for the joint application of LANs for the estimation of RL-SSM models. Given the encouraging results we plan on a number of extensions. Firstly, we incorporated LAN-based RL-SSM into the widely used HDDM python toolbox (Wiecki et al., 2013), a step towards easy community access. This addition to HDDM is also user-augmentable, so that researchers can test and ultimately share their own versions of RL-SSM models with the community through HDDM. And as a next step, it also naturally lends itself to testing hypotheses regarding how neural dynamics correlate with learning or decision parameters, via HDDMnnRegression (Fengler et al., 2021). Second, we plan to test our approach on a much larger bank of models ourselves. In the context of RL-SSM, we can e.g. test n-choice multi-armed bandits, SSMs with dynamic decision bounds, SSMs with non-gaussian noise, opening the possibility for richer tests of theoretical models that can be informed by neural dynamics. Lastly, on the RL side, we will also expand the learning rules that can be immediately combined with SSMs, one goal being to determine how these improve or hinder parameter identifiability for various combinations of RL and SSM agents.

References

Cisek, P., Puskas, G. A., & El-Murr, S. (2009). Decisions in changing conditions: the urgency-gating model. *Journal of Neuroscience*, *29*(37), 11560–11571.

Fengler, A., Govindarajan, L. N., Chen, T., & Frank, M. J. (2021). Likelihood approximation networks (lans) for fast inference of simulation models in cognitive neuroscience. *Elife*, *10*, e65074.

Fontanesi, L. (2021, February). *laurafontanesi/rlssm: First dev release.* Zenodo. Retrieved from `https://doi.org/10.5281/zenodo.4562217` doi: 10.5281/zenodo.4562217

Fontanesi, L., Gluth, S., Spektor, M. S., & Rieskamp, J. (2019). A reinforcement learning diffusion decision model for value-based decisions. *Psychonomic bulletin & review*, *26*(4), 1099–1121.

Gutmann, M. U., Dutta, R., Kaski, S., & Corander, J. (2018). Likelihood-free inference via classification. *Statistics and Computing*, *28*(2), 411–425.

Lueckmann, J.-M., Bassetto, G., Karaletsos, T., & Macke, J. H. (2019). Likelihood-free inference with emulator networks. In *Symposium on advances in approximate bayesian inference* (pp. 32–53).

Navarro, D. J., & Fuss, I. G. (2009). Fast and accurate calculations for first-passage times in wiener diffusion models. *Journal of mathematical psychology*, *53*(4), 222–230.

Neal, R. M. (2003). Slice sampling. *The Annals of Statistics*, *31*(3), 705 – 767. Retrieved from `https://doi.org/10.1214/aos/1056562461` doi: 10.1214/aos/1056562461

Papamakarios, G., & Murray, I. (2016). Fast ε-free inference of simulation models with bayesian conditional density estimation. In *Advances in neural information processing systems* (pp. 1028–1036).

Papamakarios, G., Nalisnick, E., Rezende, D. J., Mohamed, S., & Lakshminarayanan, B. (2019). Normalizing flows for probabilistic modeling and inference. *arXiv preprint*, *arXiv:1912.02762*.

Papamakarios, G., Sterratt, D., & Murray, I. (2019). Sequential neural likelihood: Fast likelihood-free inference with autoregressive flows. In *The 22nd international conference on artificial intelligence and statistics* (pp. 837–848).

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., … Chintala, S. (2019). Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, & R. Garnett (Eds.), *Advances in neural information processing systems 32* (pp. 8024–8035). Curran Associates, Inc.

Pedersen, M. L., & Frank, M. J. (2020). Simultaneous hierarchical bayesian parameter estimation for reinforcement learning and drift diffusion models: a tutorial and links to neural data. *Computational Brain & Behavior*, *3*, 458–471.

Pedersen, M. L., Frank, M. J., & Biele, G. (2017). The drift diffusion model as the choice rule in reinforcement learning. *Psychonomic bulletin & review*, *24*(4), 1234–1251.

Radev, S. T., Mertens, U. K., Voss, A., & Köthe, U. (2020). Towards end-to-end likelihood-free inference with convolutional neural networks. *British Journal of Mathematical and Statistical Psychology*, *73*(1), 23–43.

Ratcliff, R., & McKoon, G. (2008). The diffusion decision model: theory and data for two-choice decision tasks. *Neural computation*, *20*(4), 873–922.

Rescorla, R. A. (1972). A theory of pavlovian conditioning: Variations in the effectiveness of reinforcement and nonreinforcement. *Current research and theory*, 64–99.

Usher, M., & McClelland, J. L. (2001). The time course of perceptual choice: the leaky, competing accumulator model. *Psychological review*, *108*(3), 550.

Wiecki, T. V., Sofer, I., & Frank, M. J. (2013). Hddm: Hierarchical bayesian estimation of the drift-diffusion model in python. *Frontiers in neuroinformatics*, *7*, 14.

Wieschen, E. M., Voss, A., & Radev, S. (2020). Jumping to conclusion? a lévy flight model of decision making. *The Quantitative Methods for Psychology*, *16*(2), 120–132.