Approximate Thompson Sampling via Epistemic Neural Networks (Supplementary Material)

Ian Osband[1] **Zheng Wen**[1] **Seyed Mohammad Asghari**[1] **Vikranth Dwaracherla**[1]
Morteza Ibrahimi[1] **Xiuyuan Lu**[1] **Benjamin Van Roy**[1]

[1]Efficient Agent Team, DeepMind , Mountain View, CA

The *Behaviour Suite for Core Reinforcement Learning* [**?**], or `bsuite` for short, is a collection of carefully-designed experiments that investigate core capabilities of a reinforcement learning (RL) agent. The aim of the `bsuite` project is to collect clear, informative and scalable problems that capture key issues in the design of efficient and general learning algorithms and study agent behaviour through their performance on these shared benchmarks. We test agents which use ENNs to represent uncertainty in action-value functions.

0.1 AGENT DEFINITION

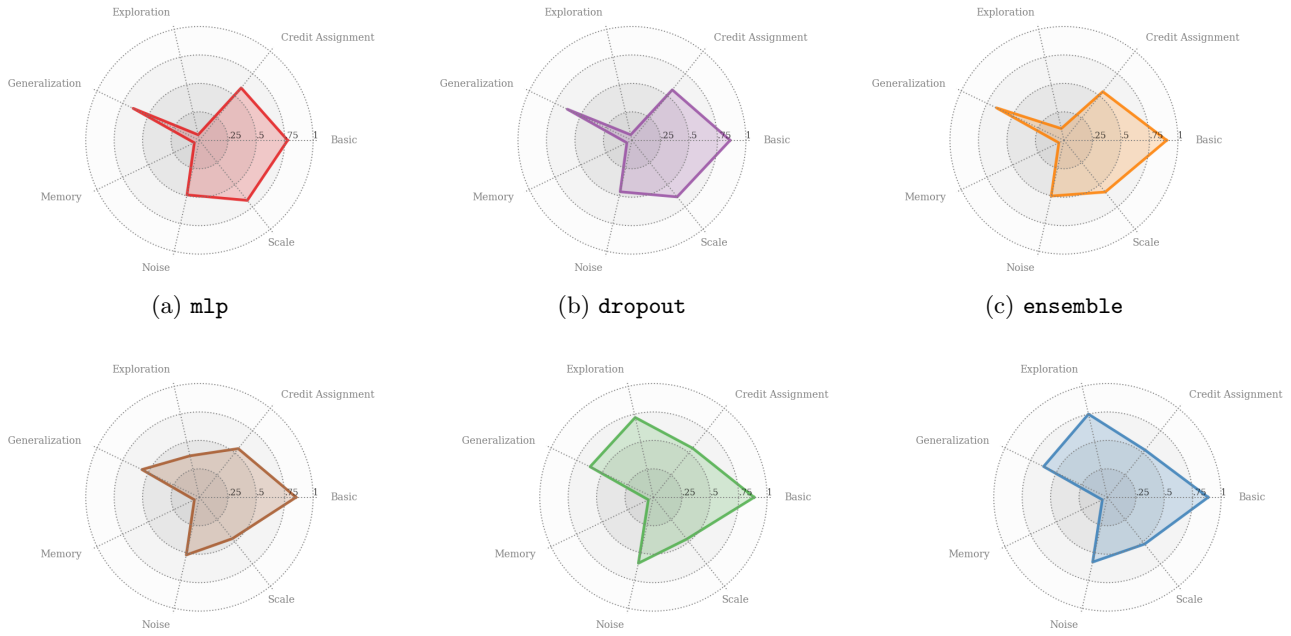
In these experiments we use the DQN variants defined in `enn_acme/experiments/bsuite`. These agents differ principally in terms of their ENN definition, which are taken directly from the `neural_testbed/agents/factories` as tuned on the Neural Testbed. We provide a brief summary of the ENNs used by agents:

- `mlp`: A 'classic' DQN network with 2-layer MLP.
- `ensemble`: An ensemble of DQN networks which only differ in initialization.
- `dropout`: An MLP with dropout used as ENN [**?**].
- `hypermodel`: A linear hypermodel [**?**].
- `ensemble+`: An ensemble of DQN networks with additive prior [**??**].
- `epinet`: The epinet architecture from **?**, reviewed in Section **??**.

0.2 SUMMARY SCORES

Each `bsuite` experiment outputs a summary score in [0,1]. We aggregate these scores by according to key experiment type, according to the standard analysis notebook.

Accepted for the 39th Conference on Uncertainty in Artificial Intelligence (UAI 2023).



(a) `mlp` (b) `dropout` (c) `ensemble`

(d) `hypermodel` (e) `epinet` (f) `ensemble+`

Figure 1: Radar plots give a snapshot of agent capabilities.

0.3 RESULTS COMMENTARY

- `mlp` performs well on basic tasks, and quite well on credit assignment, generalization, noise and scale. However, DQN performs extremely poorly across memory and exploration tasks. Our results match the high-level performance of the `bsuite/baselines`.
- `ensemble` performs similar to `mlp` agent. The additional diversity provided by random initialization in ensemble particles is insufficient to drive significantly different behaviour.
- `dropout` performs very similar to `mlp` agent. Different dropout masks are not sufficient to drive significantly different behaviour on `bsuite`.
- `hypermodel` performs better than `mlp`, `ensemble`, and `dropout` agents on exploration tasks, but the performance does not scale to the most challenging tasks in `bsuite`.
- `ensemble+` also known as Bootstrapped DQN [**??**]. Mostly performs similar to `ensemble` agent, except for exploration where it greatly outperforms `mlp`, `ensemble`, and `dropout` agents. The addition of prior functions is crucial to this performance.
- `epinet` performs similar to `ensemble+` agent, but with much lower compute. We do see some evidence that, compared to other approaches `epinet` agent is less robust to problem *scale*. This matches our observation in supervised learning that epinet performance is somewhat sensitive to the chosen scaling of the prior networks σ^P.

None of the agents we consider have a mechanism for memory as they use feed-forward networks. We could incorporate memory by considering modifications to the agents, but we don't explore that here.

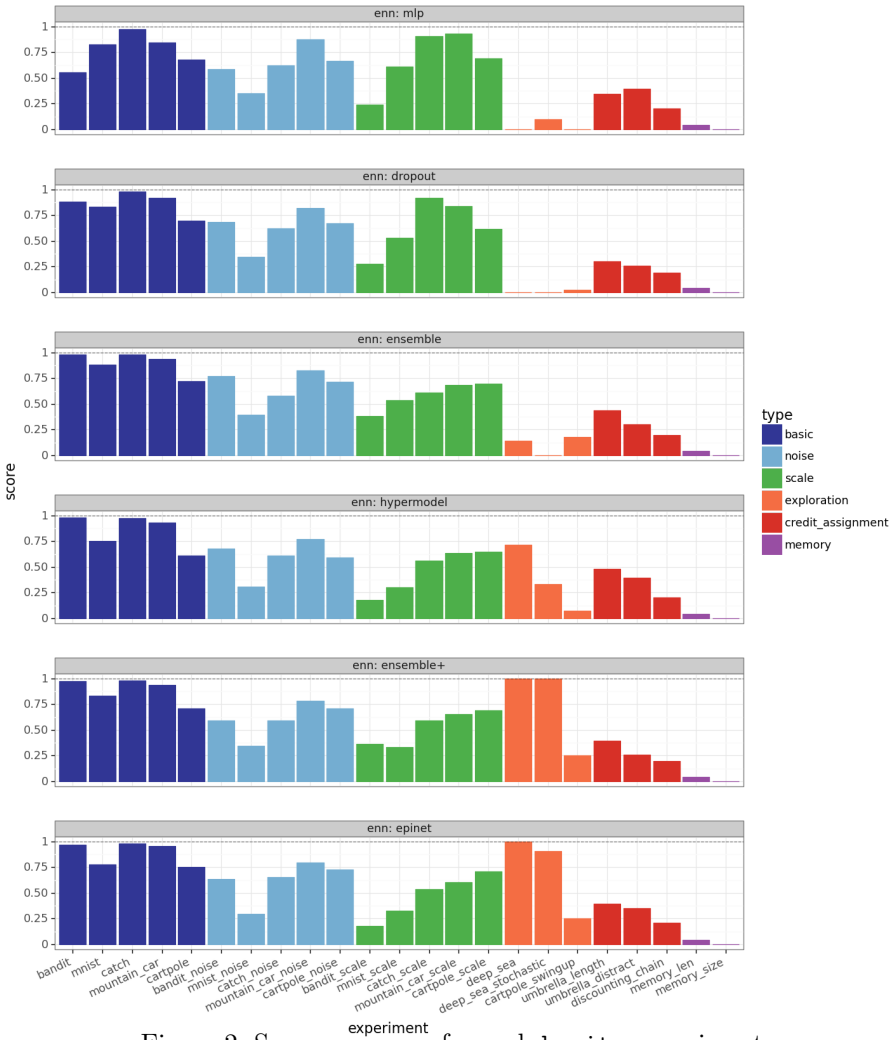


Figure 2: Summary score for each `bsuite` experiment.