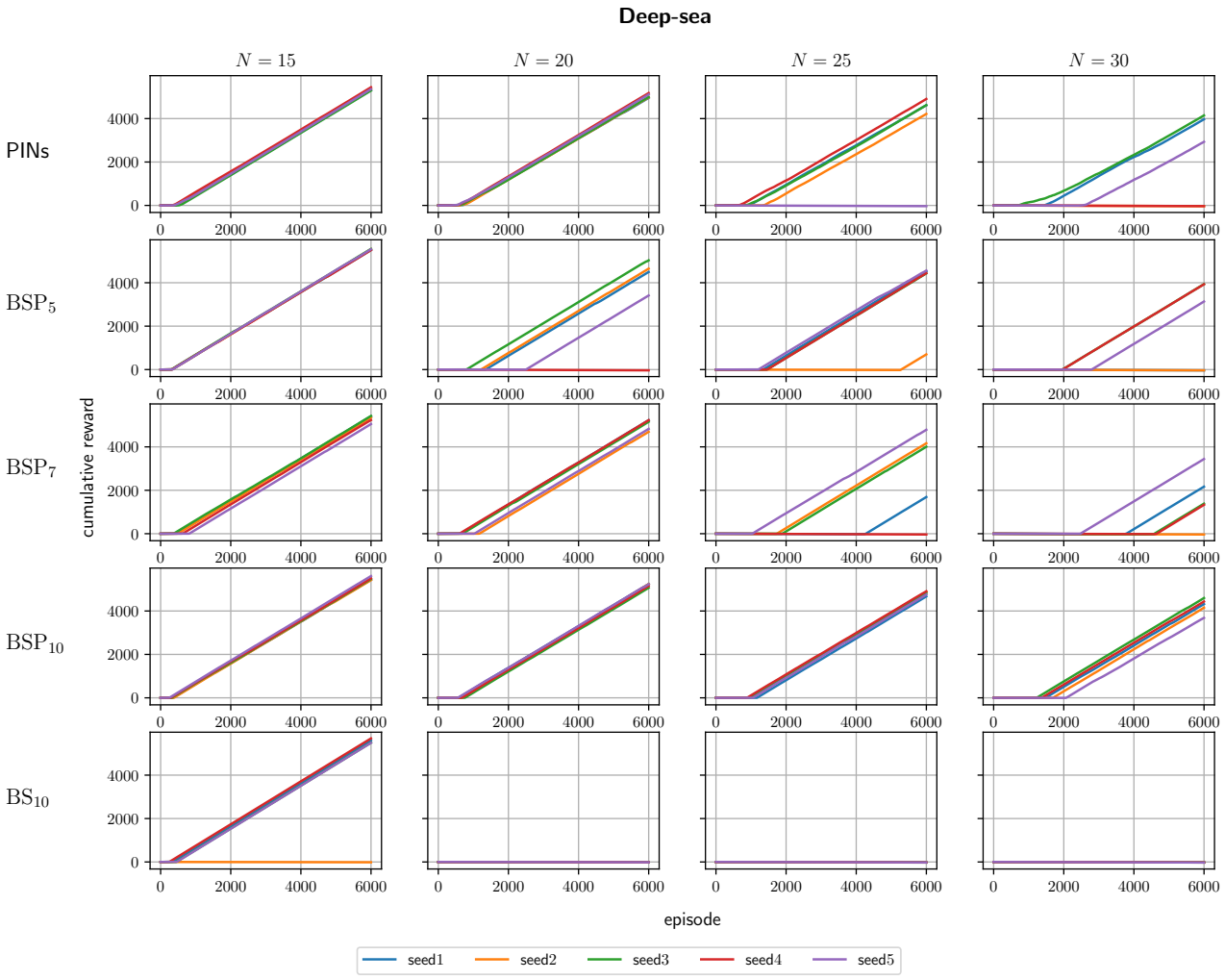

Figure 2: Comparison on cumulative reward with different problem size Ns among PINs, BSP_5, BSP_7, BSP_{10} and BS_{10}

to each output head so that data in buffer is not completely shared among heads (**?**), and (2) each head is paired and trained together with a slightly different prior $\bar{m}^u_{\bar{\omega}}(s,a)$.

For each episode l, we first sample an index $z^l \sim \mathcal{N}(0,1)$ and an output head $u \sim \text{Unif}(\{1,\dots,U\})$. The agent then acts greedily with respect to the sampled value function with *additive* priors $Q^u_{z^l} = \nu + m^u z^l + \beta_1 \bar{\nu} + \beta_2 \bar{m}^u z^l$ for consistent exploration in episode l, where β_1, β_2 are scaling hyperparameters for the mean prior and the uncertainty prior, respectively. Here the additive prior distribution can be seen as $\overline{Q}^u_{z^l} = \beta_1 \bar{\nu} + \beta_2 \bar{m}^u z^l$. A detailed training algorithm for PINs is included in supplemental material A.

5 Experimental Results

We evaluate the performance of PINs on two benchmark problems, *Deep-sea* and *Cartpole Swing-up*, that highlight the need for deep exploration from Deepmind *bsuite* (**?**), and compare it with the state-of-the-art ensemble sampling methods: the bootstrapped DQN with additive prior networks (BSP_K) (**?**) and the bootstrapped DQN without prior mechanism (BS_K) (**?**), where K denotes the number of networks in the ensemble.

Deep-sea

Deep-sea is a family of grid-like deterministic environments (**?**), which are indexed by a problem size $N \in \mathbb{N}$, with $N \times N$ cells as states, and sampled action mask $M_{ij} \sim \text{Ber}(0.5)$, $i,j \in \{1,\dots,N\}$. Action set $\mathcal{A} = \{0,1\}$, and at cell (i,j), M_{ij} represents action "left" and $1 - M_{ij}$ represents action "right". The agent always starts in the upper-left-most cell at the beginning of each episode. At each cell, action "left" ("right") takes the agent to the cell immediately to the left (right) and below. Thus, each episode lasts exactly N time steps and the agent can never revisit the same state within an episode. No cost or reward is associated with action "left". However, taking action "right" results in a cost of $0.01/N$ in cells along the main diagonal except the lower-right-most cell where a reward of 1 is given for taking action "right". Therefore, the optimal policy is picking action "right" at

each step giving an episodic reward of 0.99. All other policies generate zero or negative rewards. The usual dithering methods will need $\Omega\left(2^N\right)$ episodes to learn the optimal policy, which grows exponentially with the problem size N.

Figure 2 shows the cumulative reward of our PINs and various ensemble models on Deep-sea with four different sizes $N = 15, 20, 25, 30$ for $6K$ episodes of learning. Each approach is evaluated over 5 different random seeds. We consider that the agent has learned the optimal policy when there is a linear increase in cumulative reward by the end of training. For example, our agent with PINs successfully learned the optimal policy in $3/5$ seeds when $N = 30$, in $4/5$ seeds when $N = 25$ within $6K$ episodes. All networks are MLP with 1 hidden layer. For PINs, the mean network has 300 units and the uncertainty network has 512 units with $U = 10$ output heads in the output layer. We set $\sigma = 2$ for the added noise without any decay for experiments on Deep-sea, and $\beta_1 = \beta_2 = 2$. For ensemble models (**?**), each single network in the ensemble contains 50 hidden units, and we set prior scale $\beta = 10$ for BSP as recommended in (**?**). For BS, we simply let $\beta = 0$ to exclude the prior mechanism. For all bootstrapping, we use Bernoulli mask with $p = 0.5$. We see that the performance of PINs is comparable to that of ensemble methods BSP_5 and BSP_7 with additive priors. Also, note that the PINs are relatively more efficient in computation as PINs only require 2 back-propagations per update while BSP_K need K backward passes per update. In addition, even equipped with 10 separate networks, BS_{10} struggles to learn the optimal policy as N increases, which highlights the significance of a prior for efficient exploration.

Conceptually, the main advantage of our PINs is that it distributes the tasks of learning an value function and measuring uncertainty in estimates into two separate networks. Therefore, it is possible to further enhance exploration by using a more carefully-crafted and more complex design of the uncertainty network and its prior without concerning about the stability of learning in the mean network. As an example, a more delicate design that induces diverse uncertainty estimates for unseen state-action pairs can potentially drive exploration to the next-level. We consider experimenting with different architectures of the uncertainty network as our future work.

Cartpole Swing-up

We next present our results on a classic benchmark problem: *Cartpole Swing-up* (**?**), (**?**), which requires learning a more complex mapping from continuous states[1] to action values. Unlike the original cartpole, the problem is modified so that the pole is hanging down initially and the agent only receives a reward of 1 when the pole is nearly upright, balanced, and centered[2] Further, a cost of 0.05 is added to move the cart and the environment is simulated under timescale

Figure 3 shows a comparison on smoothed episodic reward in 3000 episodes of learning over 5 different random seeds. We failed to train a BS_{10} agent that can learn a performant policy on this environment; thus, the results are omitted. To demonstrate computational savings of PINs, all networks used here have 3 hidden layers with 50 units in each layer. Besides, the uncertainty network spins out only $U = 2$ output heads. For added noise in PINs, we use $\sigma = 2$ and linearly decay it to 1 over the course of training to promote concentration in the approximate posterior distribution. As for prior scale, we use $\beta_1 = \beta_2 = 2$ for our PINs, and $\beta = 30$ for BSP as in (**?**). We see that PINs achieved similar performance to that of the ensemble models but with only two separate neural networks. Additionally, although PINs seem to progress slowly compared to BSP_K, they exhibit smaller variance in performance by the end of learning. This experiment demonstrates the computational efficiency that can be brought by PINs and by index sampling for

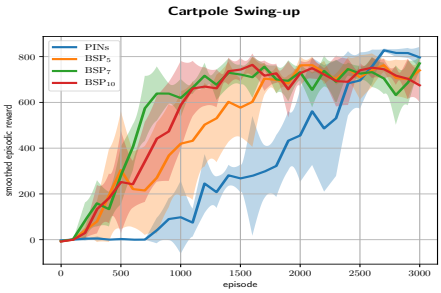

Figure 3: Comparison on smoothed episodic reward: each point is the maximum episodic reward within the most recent 100 episodes. We plot the average performance over 5 random seeds for each method and the shaded area represents $+/-$ standard deviation.

6 Conclusion

In this paper, we present a parameterized indexed value function which can be learned by a distributional version of TD. After proving its efficiency in the tabular setting, we introduce a computationally lightweight dual-network architecture, Parameterized Indexed Networks (PINs), for deep RL and show its efficacy through numerical experiments. To the best of our knowledge, we lead the first study of index sampling to achieve efficient exploration in the field of RL.

However, several open questions are still left unanswered. It would be worthwhile to explore other designs for uncertainty and prior networks, and experiment with other distributional metrics to see if one can obtain stronger theoretical guarantees and/or better empirical performance. It would be very7 interesting to combine the ideas of PINs with more advanced neural architectures such as convolutional networks, and evaluate its performance on Atari games with sparse rewards. We leave this and many possible extensions

7 Acknowledgement

We thank Benjamin Van Roy, Chengshu Li for the insightful

[1] A 8 dimensional vector in *bsuite* where we set the threshold of position x to be 5.

[2] Reward 1 only when $\cos(\theta) > 0.95, |x| < 1, |\dot{x}| < 1$ and $|\dot{\theta}| < 1$.