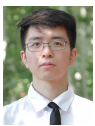
HyperDQN: A Randomized Exploration for Deep Reinforcement Learning

Yingru Li
yingruli@link.cuhk.edu.cn

NeurIPS 2021 Workshop on Ecological Theory of RL (Oral)





Ziniu Li[1] Yingru Li[1,†] Yushun Zhang[1] Tong Zhang[2] Zhi-Quan Luo[1]

[1] The Chinese University of Hong Kong, Shenzhen [2] Hong Kong University of Science and Technology

[†] Corresponding author

Contributions

- We present a practical randomized exploration method *HyperDQN*.
- Our experiments support that HyperDQN achieves significant improvements.
  - HyperDQN achieves about 2x improvement than baselines over 56 tasks in Atari suite.
  - HyperDQN outperforms all baselines on 7 out of 9 tasks in SuperMarioBros Games.

Contributions

- We present a practical randomized exploration method *HyperDQN*.

- Our experiments support that HyperDQN achieves significant improvements.
  - HyperDQN achieves about 2x improvement than baselines over 56 tasks in Atari suite.
  - HyperDQN outperforms all baselines on 7 out of 9 tasks in SuperMarioBros Games.

Contributions

- We present a practical randomized exploration method *HyperDQN*.

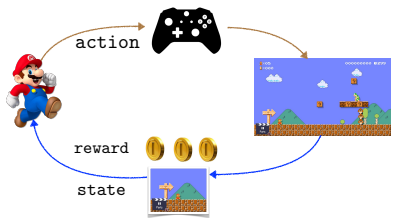
- Our experiments support that HyperDQN achieves significant improvements.
  - HyperDQN achieves about 2x improvement than baselines over 56 tasks in Atari suite.
  - HyperDQN outperforms all baselines on 7 out of 9 tasks in SuperMarioBros Games.

Outline

Reinforcement Learning

- An RL agent interacts with an MDP $\mathcal{M} = (\mathcal{S}, \mathcal{A}, P, R, \gamma)$ to maximize cumulative reward.

$$\max_{\pi} \mathbb{E}_{\pi} \left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) \right].$$



Exploration in RL

- **A fundamental question in RL:** the exploration-exploitation trade-off.
  - Exploration: explore highly uncertain states and actions, which may sacrifice immediate reward.
  - Exploitation: take the best-known action, which may be sub-optimal due to partial information.
- We aim to design efficient exploration strategies in this work.

Exploration in RL

- **A fundamental question in RL:** the exploration-exploitation trade-off.
  - Exploration: explore highly uncertain states and actions, which may sacrifice immediate reward.
  - Exploitation: take the best-known action, which may be sub-optimal due to partial information.
- We aim to design efficient exploration strategies in this work.

Existing Methods for Exploration

Three types of exploration methods:

- **Dithering strategies**:
  epsilon-greedy [Mnih et al., 2015], Gaussian noise [Lillicrap et al., 2016].
- **Exploration bonus based exploration**:
  UCB and its variants [Stadie et al., 2015, Pathak et al., 2017, Tang et al., 2017, Burda et al., 2019, Bai et al., 2021].
- **Randomized exploration**:
  RLSVI [Osband et al., 2016b] and BootDQN [Osband et al., 2016a].

We will discuss randomized exploration, particularly RLSVI.

Existing Methods for Exploration

Three types of exploration methods:

- **Dithering strategies**:
  epsilon-greedy [Mnih et al., 2015], Gaussian noise [Lillicrap et al., 2016].
- **Exploration bonus based exploration**:
  UCB and its variants [Stadie et al., 2015, Pathak et al., 2017, Tang et al., 2017, Burda et al., 2019, Bai et al., 2021].
- **Randomized exploration**:
  RLSVI [Osband et al., 2016b] and BootDQN [Osband et al., 2016a].

We will discuss randomized exploration, particularly RLSVI.

Existing Methods for Exploration

Three types of exploration methods:

- **Dithering strategies**:
  epsilon-greedy [Mnih et al., 2015], Gaussian noise [Lillicrap et al., 2016].
- **Exploration bonus based exploration**:
  UCB and its variants [Stadie et al., 2015, Pathak et al., 2017, Tang et al., 2017, Burda et al., 2019, Bai et al., 2021].
- **Randomized exploration**:
  RLSVI [Osband et al., 2016b] and BootDQN [Osband et al., 2016a].

We will discuss randomized exploration, particularly RLSVI.

Existing Methods for Exploration

Three types of exploration methods:

- **Dithering strategies**:
  epsilon-greedy [Mnih et al., 2015], Gaussian noise [Lillicrap et al., 2016].
- **Exploration bonus based exploration**:
  UCB and its variants [Stadie et al., 2015, Pathak et al., 2017, Tang et al., 2017, Burda et al., 2019, Bai et al., 2021].
- **Randomized exploration**:
  RLSVI [Osband et al., 2016b] and BootDQN [Osband et al., 2016a].

We will discuss randomized exploration, particularly RLSVI.

Review: RLSVI

Randomized Least-Square Value Iteration (RLSVI) [Osband et al., 2016b].

- (Step 1) Sample model parameters $\tilde{\theta}$ from **posterior distribution of** θ^*.
- (Step 2) For each stage t, take greedy action: $a_t = \text{argmax}_a Q(s_t, a)$, where $Q(s_t, a) := \phi(s_t, a)^\top \tilde{\theta}$.
- (Step 3) (Key step) Update posterior distribution of θ^*.
  - When feature ϕ is fixed and known, posterior update is computational friendly.

However, we observe that Step 3 is intractable in Deep RL. We elaborate as follows.

Review: RLSVI

Randomized Least-Square Value Iteration (RLSVI) [Osband et al., 2016b].

- (Step 1) Sample model parameters $\tilde{\theta}$ from **posterior distribution of** θ^*.
- (Step 2) For each stage t, take greedy action: $a_t = \text{argmax}_a \, Q(s_t, a)$, where $Q(s_t, a) := \phi(s_t, a)^\top \tilde{\theta}$.
- (Step 3) (Key step) Update posterior distribution of θ^*.
  - When feature ϕ is fixed and known, posterior update is computational friendly.

However, we observe that Step 3 is intractable in Deep RL. We elaborate as follows.

Review: RLSVI

Randomized Least-Square Value Iteration (RLSVI) [Osband et al., 2016b].

- (Step 1) Sample model parameters $\tilde{\theta}$ from **posterior distribution of** θ^*.
- (Step 2) For each stage t, take greedy action: $a_t = \text{argmax}_a Q(s_t, a)$, where $Q(s_t, a) := \phi(s_t, a)^\top \tilde{\theta}$.
- (Step 3) (Key step) Update posterior distribution of θ^*.
  - When feature ϕ is fixed and known, posterior update is computational friendly.

However, we observe that Step 3 is intractable in Deep RL. We elaborate as follows.

Review: RLSVI

Randomized Least-Square Value Iteration (RLSVI) [Osband et al., 2016b].

- (Step 1) Sample model parameters $\tilde{\theta}$ from **posterior distribution of** θ^*.
- (Step 2) For each stage t, take greedy action: $a_t = \text{argmax}_a \, Q(s_t, a)$, where $Q(s_t, a) := \phi(s_t, a)^\top \tilde{\theta}$.
- (Step 3) (Key step) Update posterior distribution of θ^*.
  - When feature ϕ is fixed and known, posterior update is computational friendly.

However, we observe that Step 3 is intractable in Deep RL. We elaborate as follows.

Review: RLSVI

Step 3 of RLSVI: at episode K, we need to update the **posterior covariance**:

$$\mathrm{Cov}[\theta^* \mid \mathcal{D}] = \left(\frac{1}{\sigma_\omega^2}\Phi_K + \frac{1}{\sigma_p^2}I\right)^{-1}, \quad \Phi_K = \sum_{k=1}^{K} \phi(s_k, a_k)\phi(s_k, a_k)^\top \in \mathbb{R}^{d\times d}. \tag{1}$$

When extending to Deep RL, we observe two issues:

- (Issue 1) RLSVI assumes a good feature ϕ is known and fixed in advance.
- (Issue 2) When ϕ is changing, $\mathrm{Cov}[\theta^* \mid \mathcal{D}]$ cannot be computed efficiently.

Review: RLSVI

Step 3 of RLSVI: at episode K, we need to update the **posterior covariance**:

$$\mathrm{Cov}[\theta^* \mid \mathcal{D}] = \left(\frac{1}{\sigma_\omega^2} \Phi_K + \frac{1}{\sigma_p^2} I \right)^{-1}, \quad \Phi_K = \sum_{k=1}^{K} \phi(s_k, a_k) \phi(s_k, a_k)^\top \in \mathbb{R}^{d \times d}. \tag{1}$$

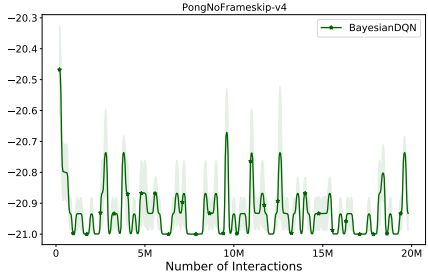
When extending to Deep RL, we observe two issues:

- (Issue 1) RLSVI assumes a good feature ϕ is known and fixed in advance.
- (Issue 2) When ϕ is changing, $\mathrm{Cov}[\theta^* \mid \mathcal{D}]$ cannot be computed efficiently.

Challenges in Deep RL

(Issue 1) RLSVI assumes a good feature ϕ is known and fixed in advance.

- In Deep RL: Good features are unknown and need to be learned.
- Without good features, the performance of RLSVI (Bayesian DQN [Azizzadenesheli et al., 2018]) is poor in Deep RL.



Challenges in Deep RL

(Issue 2) when ϕ is changing, $\mathrm{Cov}[\theta^* \mid \mathcal{D}]$ cannot be computed efficiently.

- Assume $\sigma_p = \sigma_w = 1$ in Equation (1) and denote (s_K, a_K) by x_K.

$$\text{fixed } \phi\text{:} \quad \Phi_K = \Phi_{K-1} + \phi(x_K)\phi(x_K)^\top \quad \text{with } \Phi_0 = \boldsymbol{I},$$

$$\text{changing } \phi_K\text{:} \quad \Phi_K := \sum_{\ell=1}^{K} \phi_K(x_\ell)\phi_K(x_\ell)^\top,\ \Phi_{K-1} := \sum_{\ell=1}^{K-1} \phi_{K-1}(x_\ell)\phi_{K-1}(x_\ell)^\top, \cdots$$

Challenges in Deep RL

(Issue 2) when ϕ is changing, $\mathrm{Cov}[\theta^* \mid \mathcal{D}]$ cannot be computed efficiently.

- Assume $\sigma_p = \sigma_w = 1$ in Equation (1) and denote (s_K, a_K) by x_K.

$$\text{fixed } \phi: \quad \Phi_K = \Phi_{K-1} + \phi(x_K)\phi(x_K)^\top \quad \text{with } \Phi_0 = \boldsymbol{I},$$

$$\text{changing } \phi_K: \quad \Phi_K := \sum_{\ell=1}^{K} \phi_K(x_\ell)\phi_K(x_\ell)^\top, \; \Phi_{K-1} := \sum_{\ell=1}^{K-1} \phi_{K-1}(x_\ell)\phi_{K-1}(x_\ell)^\top, \cdots$$

Challenges in Deep RL

(Issue 2) when ϕ is changing, $\mathrm{Cov}[\theta^* \mid \mathcal{D}]$ cannot be computed efficiently.

- Assume $\sigma_p = \sigma_w = 1$ in Equation (1) and denote (s_K, a_K) by x_K.

$$\text{fixed } \phi: \quad \Phi_K = \Phi_{K-1} + \phi(x_K)\phi(x_K)^\top \quad \text{with } \Phi_0 = \boldsymbol{I},$$

$$\text{changing } \phi_K: \quad \Phi_K := \sum_{\ell=1}^{K} \phi_K(x_\ell)\phi_K(x_\ell)^\top, \ \Phi_{K-1} := \sum_{\ell=1}^{K-1} \phi_{K-1}(x_\ell)\phi_{K-1}(x_\ell)^\top, \cdots$$

Challenges in Deep RL

(Issue 2) when ϕ is changing, $\mathrm{Cov}[\theta^* \mid \mathcal{D}]$ cannot be computed efficiently.

- Assume $\sigma_p = \sigma_w = 1$ in Equation (1) and denote (s_K, a_K) by x_K.

$$\text{fixed } \phi: \quad \Phi_K = \Phi_{K-1} + \phi(x_K)\phi(x_K)^\top \quad \text{with } \Phi_0 = \boldsymbol{I},$$

$$\text{changing } \phi_K: \quad \Phi_K := \sum_{\ell=1}^{K} \phi_K(x_\ell)\phi_K(x_\ell)^\top, \ \Phi_{K-1} := \sum_{\ell=1}^{K-1} \phi_{K-1}(x_\ell)\phi_{K-1}(x_\ell)^\top, \cdots$$

- In the changing ϕ_K case, Φ_K has to be recomputed using all historical data.
  - e.g. in Atari, this calculation could involve more than 1M samples with dimension 512.
- Furthermore, we need to inverse Φ_k in Equation (1).

Challenges in Deep RL

To tackle (Issue 1) & (Issue 2) in updating the posterior distribution of θ^*.

- BootDQN [Osband et al., 2016a] uses **ensembles** to approximate the posterior.
  - But the number of ensembles is often limited $\rightarrow$ poor approximation.

- In this work, we introduce *HyperDQN*, which addresses the above issues in Deep RL.

Challenges in Deep RL

To tackle (Issue 1) & (Issue 2) in updating the posterior distribution of θ^*.

- BootDQN [Osband et al., 2016a] uses **ensembles** to approximate the posterior.
  - But the number of ensembles is often limited $\rightarrow$ poor approximation.

- In this work, we introduce *HyperDQN*, which addresses the above issues in Deep RL.

Outline

Outline

Overview of HyperDQN

Two models are implemented in HyperDQN.

- Base model: DQN-type structure

$$Q_\theta(s,a) = \langle \phi_{\theta_{\text{hidden}}}(s), \theta_{\text{predict}}(a) \rangle.$$

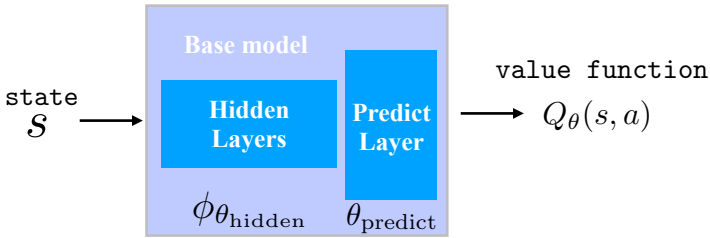


Figure 1: Illustration for the proposed method HyperDQN: Base model.

Overview of HyperDQN

Two models are implemented in HyperDQN.

- Base model: DQN-type structure $Q_\theta(s,a) = \langle \phi_{\theta_{\text{hidden}}}(s), \theta_{\text{predict}}(a) \rangle$.
- Hypermodel [Dwaracherla et al., 2020]: $\theta_{\text{predict}} = f_\nu(z)$ where $z \sim p(z)$.
- Resulting model: $Q_{\theta_{\text{hidden}}, f_\nu(z)}(s,a)$.

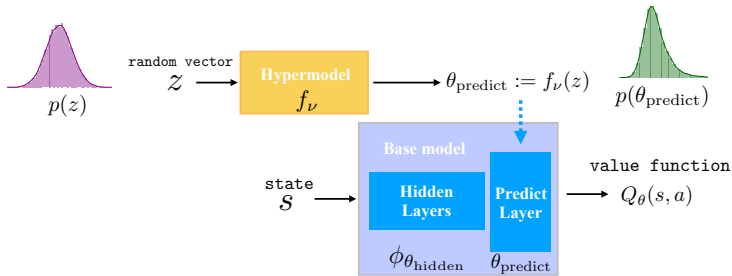


Figure 2: Illustration for the proposed method HyperDQN.

Outline

Training Objective

Training objective in HyperDQN:

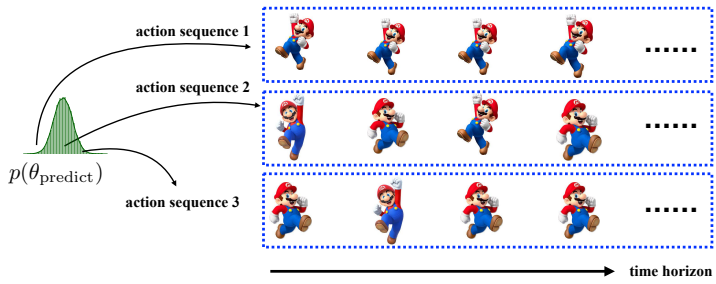
$$\min_{\nu,\theta_{\text{hidden}}} \int_z p(z) \Bigg[\sum_{(s,a,r,\xi,s')\in\mathcal{D}} \left(Q_{\text{target}}(s',z) + \sigma_\omega z^\top \xi - Q_{\text{prediction}}(s,a,z)\right)^2 + \frac{\sigma_\omega^2}{\sigma_p^2} \left\| f_\nu(z) - f_{\nu_{\text{prior}}}(z) \right\|^2 \Bigg] (\mathrm{d}z), \tag{2}$$

where

$$\begin{aligned} Q_{\text{prediction}}(s,a,z) &= Q_{\theta_{\text{hidden}}, f_\nu(z)}(s,a), \\ Q_{\text{target}}(s',z) &= r + \gamma \max_{a'} \left[Q_{\bar{\theta}_{\text{hidden}}, f_{\bar{\nu}}(z)}(s',a') \right]. \end{aligned} \tag{3}$$

- Noise term $\sigma_\omega z^\top \xi$ is used for posterior approximation and will be explained later.
- **Joint Feature Learning and Uncertainty quantification** through Equation (2).

Diverse Action Sequences Induced by HyperDQN



From the (approximate) posterior distribution, all plausible action sequences can be sampled for exploration using $z \sim p(z), \theta_{\textbf{predict}} = f_\nu(z)$ **and** $\operatorname{argmax}_a Q_{\theta_{\textbf{hidden}}, \theta_{\textbf{predict}}}(s, a)$.

HyperDQN Algorithm

- Compared with DQN, our method incorporates the hypermodel for randomized exploration.
  - Can be regarded as an extension of hypermodel from bandit to RL tasks.
- Importantly, there is *NO* epsilon-greedy in HyperDQN.
  - Surprisingly, many existing advanced exploration methods [Osband et al., 2016a, Rashid et al., 2020, Bai et al., 2021] rely on epsilon-greedy.
  - Without epsilon-greedy, the performance of these methods could degenerate.
- We find that using epsilon-greedy for HyperDQN ruins deep-insight behaviors and leads to a worse performance (Figure 4).

HyperDQN Algorithm

- Compared with DQN, our method incorporates the hypermodel for randomized exploration.
  - Can be regarded as an extension of hypermodel from bandit to RL tasks.
- Importantly, there is *NO* epsilon-greedy in HyperDQN.
  - Surprisingly, many existing advanced exploration methods [Osband et al., 2016a, Rashid et al., 2020, Bai et al., 2021] rely on epsilon-greedy.
  - Without epsilon-greedy, the performance of these methods could degenerate.
- We find that using epsilon-greedy for HyperDQN ruins deep-insight behaviors and leads to a worse performance (Figure 4).

HyperDQN Algorithm

- Compared with DQN, our method incorporates the hypermodel for randomized exploration.
  - Can be regarded as an extension of hypermodel from bandit to RL tasks.
- Importantly, there is *NO* epsilon-greedy in HyperDQN.
  - Surprisingly, many existing advanced exploration methods [Osband et al., 2016a, Rashid et al., 2020, Bai et al., 2021] rely on epsilon-greedy.
  - Without epsilon-greedy, the performance of these methods could degenerate.
- We find that using epsilon-greedy for HyperDQN ruins deep-insight behaviors and leads to a worse performance (Figure 4).

Outline

Atari

- OB2I [Bai et al., 2021]: a SOTA exploration bonus based method.

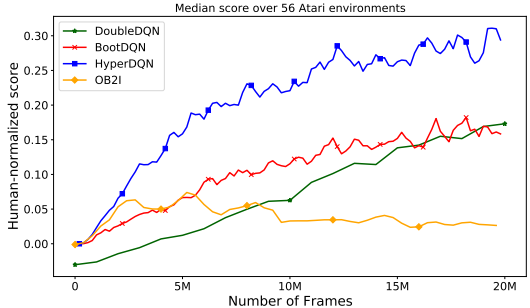


Figure 3: Human-normalized score over 56 environments in Atari 2600 suite.

HyperDQN has 2x improvement over baselines.

SuperMarioBros

Table 1: The mean evaluation scores (after 20M frames) for SuperMarioBros games.

	DoubleDQN	BootDQN	OB2I	HyperDQN
SuperMarioBros-1-1-v1	**8,698**	7,008	4,457	7,924
SuperMarioBros-1-2-v1	5,903	5,665	4,695	**8,266**
SuperMarioBros-1-3-v1	1,989	1,609	1,583	**6,046**
SuperMarioBros-2-1-v1	**31,247**	26,415	14,225	23,046
SuperMarioBros-2-2-v1	1,622	1,092	1,587	**1,983**
SuperMarioBros-2-3-v1	5,515	5,107	4,401	**5,980**
SuperMarioBros-3-1-v1	4,463	3,861	3,251	**48,384**
SuperMarioBros-3-2-v1	20,511	20,954	26,508	**41,139**
SuperMarioBros-3-3-v1	3,416	2,650	3,009	**5,568**

HyperDQN outperforms over baselines in 7/9 games.

SuperMarioBros

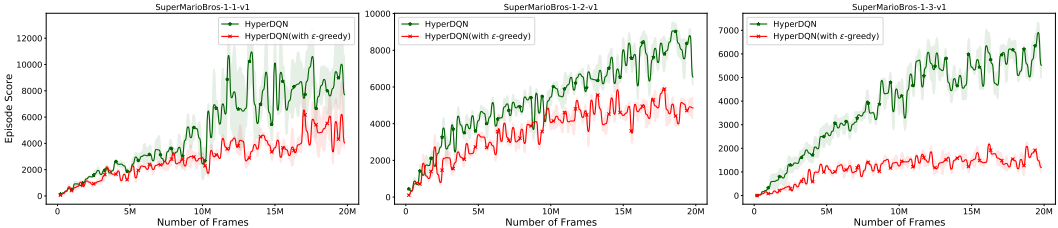


Figure 4: Ablation study about epsilon-greedy in HyperDQN.

Using epsilon-greedy ruins randomized exploration behaviors of HyperDQN.

SuperMarioBros

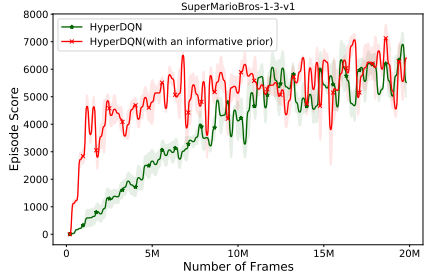


Figure 5: Ablation study about informative prior in HyperDQN.

Using an informative prior model in Objective function (2) could accelerate exploration.

Computation Efficiency

- Computation complexity comparison with BootDQN on Deep-Sea [Osband et al., 2020].
- The metric is (a smaller number indicates a better performance):

$$\textbf{computation complexity} = n_{\text{sgd}} \times n_z \times K.$$

  - n_{sgd} is the number of SGD steps per iteration
  - n_z is the number of ensemble (index) samples,
  - K is the number of episode that the episode return is 0.99.

	deep-sea-10	deep-sea-15	deep-sea-20	deep-sea-25	deep-sea-30
BootDQN	130K	250K	490K	870K	1,640K
HyperDQN	**48K**	**104K**	**196K**	**304K**	**1,120K**

Outline

Posterior Approximation Ability of Hypermodel

- [Dwaracherla et al., 2020] show that a linear hypermodel has sufficient representation power.
- However, [Dwaracherla et al., 2020] do not demonstrate why hypermodel can learn the posterior distribution.

Theorem 1 [Our Work] [Informal]

When both base & hypermodel are linear, hypermodel can generate approximate posterior samples of $\theta^\star$.

Posterior Approximation Ability of Hypermodel

- [Dwaracherla et al., 2020] show that a linear hypermodel has sufficient representation power.
- However, [Dwaracherla et al., 2020] do not demonstrate why hypermodel can learn the posterior distribution.

Theorem 1 [Our Work] [Informal]

When both base & hypermodel are linear, hypermodel can generate approximate posterior samples of $\theta^\star$.

Posterior Approximation Ability of Hypermodel

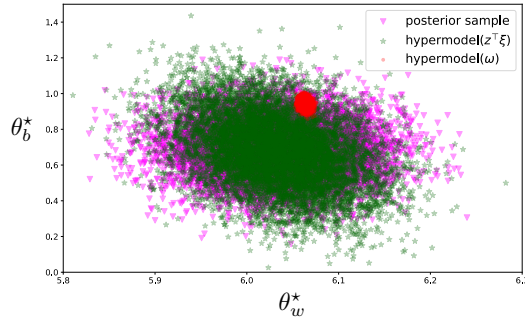


Figure 6: Visualization of true posterior samples and learned posterior samples.

Hypermodel can approximate the posterior distribution with the z-dependent noise $z^\top \xi$.

Outline

Conclusion

Summary

- Practical randomized exploration method with **strong empirical performance**.
- Provide **understanding of why** the hypermodel works.

Future Work

- Extension to continuous control tasks.
- **Informative prior** to accelerate exploration.

Conclusion

Summary

- Practical randomized exploration method with **strong empirical performance**.
- Provide **understanding of why** the hypermodel works.

Future Work

- Extension to continuous control tasks.
- **Informative prior** to accelerate exploration.

References I

K. Azizzadenesheli, E. Brunskill, and A. Anandkumar. Efficient exploration through bayesian deep q-networks. In Information Theory and Applications Workshop, pages 1–9, 2018.

C. Bai, L. Wang, L. Han, J. Hao, A. Garg, P. Liu, and Z. Wang. Principled exploration via optimistic bootstrapping and backward induction. In Proceedings of the 38th International Conference on Machine Learning, pages 577–587, 2021.

Y. Burda, H. Edwards, A. J. Storkey, and O. Klimov. Exploration by random network distillation. In Proceedings of the 7th International Conference on Learning Representations, 2019.

V. Dwaracherla, X. Lu, M. Ibrahimi, I. Osband, Z. Wen, and B. Van Roy. Hypermodels for exploration. In Proceedings of the 8th International Conference on Learning Representations, 2020.

References II

T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra. Continuous control with deep reinforcement learning. In Proceedings of the 4th International Conference on Learning Representations, 2016.

V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, et al. Human-level control through deep reinforcement learning. Nature, 518(7540):529–533, 2015.

I. Osband, C. Blundell, A. Pritzel, and B. Van Roy. Deep exploration via bootstrapped DQN. In Advances in Neural Information Processing Systems 29, pages 4026–4034, 2016a.

I. Osband, B. V. Roy, and Z. Wen. Generalization and exploration via randomized value functions. In Proceedings of the 33rd International Conference on Machine Learning, pages 2377–2386, 2016b.

References III

I. Osband, Y. Doron, M. Hessel, J. Aslanides, E. Sezener, A. Saraiva, K. McKinney, T. Lattimore, C. Szepesvári, S. Singh, B. V. Roy, R. S. Sutton, D. Silver, and H. van Hasselt. Behaviour suite for reinforcement learning. In Proceedings of the 8th International Conference on Learning Representations, 2020.

D. Pathak, P. Agrawal, A. A. Efros, and T. Darrell. Curiosity-driven exploration by self-supervised prediction. In Proceedings of the 34th International Conference on Machine Learning, pages 2778–2787, 2017.

T. Rashid, B. Peng, W. Boehmer, and S. Whiteson. Optimistic exploration even with a pessimistic initialisation. In Proceedings of the 8th International Conference on Learning Representations, 2020.

B. C. Stadie, S. Levine, and P. Abbeel. Incentivizing exploration in reinforcement learning with deep predictive models. arXiv, 1507.00814, 2015.

References IV

H. Tang, R. Houthooft, D. Foote, A. Stooke, X. Chen, Y. Duan, J. Schulman, F. D. Turck, and P. Abbeel. #exploration: A study of count-based exploration for deep reinforcement learning. In Advances in Neural Information Processing Systems 30, pages 2753–2762, 2017.