HyperAgent: A Simple, Efficient and Scalable RL framework in Complex Environment

Yingru Li

yingruli@link.cuhk.edu.cn
The Chinese University of Hong Kong, Shenzhen

January 13, 2024

Reinforcement Learning Problem

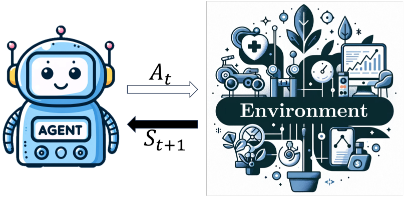


Figure: Agent-Environment Interface.
Experience: $A_0, S_1, A_1, S_2, \ldots$

Environment $M = (\mathcal{S}, \mathcal{A}, P)$

- State $S_{t+1} \sim P(\cdot \mid S_t, A_t)$.

Agent$(\mathcal{S}, \mathcal{A}, r, \mathcal{D}_t) \to \pi_t$ to max long-term rewards

- Reward $R_{t+1} = r(S_t, A_t, S_{t+1})$ where r describes the Agent's preference.
- Historical **Data** $\mathcal{D}_t = \mathcal{D}_{t-1} \cup \{A_{t-1}, S_t\}$ is **accumulated** with initial $\mathcal{D}_0 = \{S_0\}$ or $\mathcal{D}_0 = \mathcal{D}_{\text{offline}}$.
- Action $A_t \sim \pi_t(\cdot \mid S_t)$; Policy $\pi_t = \text{Agent}(\mathcal{S}, \mathcal{A}, r, \mathcal{D}_t)$ adapted to the **accumulated** $\mathcal{D}_t$.
- **Objective**: $\pi_{\text{agent}} = (\pi_0, \pi_1, \ldots)$ to maximize

$$\mathbb{E}\left[\sum_{t=0}^{T-1} R_{t+1} \mid \pi_{\text{agent}}, M\right]. \tag{1}$$

Complex Environment has Exponentially Large $\mathcal{S}$ ↑

Figure: Real-world Environment is **Complex**: **Large state space**, Non-stationary dynamics, etc

Environment $M = (\mathcal{S}, \mathcal{A}, P)$

- State $S_{t+1} \sim P(\cdot \mid S_t, A_t)$.

- Games: **Exponentially large state space** (e.g., Go $> 10^{170}$, Atari games $> 128^{(160\times 192)}$ (Raw pixels), etc.)

- Real-world applications: **High-dimensional state space** (e.g., image, video, audio, text, high-dimensional feature vectors, etc.)
  - **Healthcare**: **Patient state** (e.g., blood pressure, heart rate, health record ...)
  - **Chatbot (GPTs)**: **Conversation state** (e.g., prompt, dialogue history, accessible relevant information etc.)
  - **Communication, Robotics, Agriculture**
  - ...

Unbouned resource requirement as $\mathcal{D} \uparrow$ and $\mathcal{S} \uparrow$

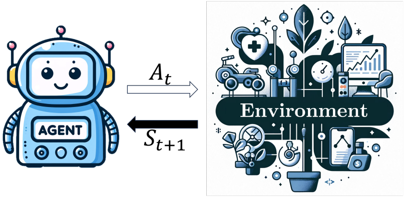


Figure: Agent-Environment Interface.
Experience: $A_0, S_1, A_1, S_2, \ldots$; and $|\mathcal{D}| \uparrow \infty$

Agent$(\mathcal{S}, \mathcal{A}, r, \mathcal{D}_t) \rightarrow \pi_t$ to max long-term rewards

- Policy $\pi_t = \text{Agent}(\mathcal{S}, \mathcal{A}, r, \mathcal{D}_t)$ adapted to **accumulated** $\mathcal{D}_t$ with size $\uparrow$ and taking **large** $\mathcal{S}$ as input.
- Resource constraints on **memory** and **computation**.
- **NOT tractable** to retrain the entire history data $\mathcal{D}$ from scratch; otherwise memory and computation requirement **growing unbounded** as $|\mathcal{D}| \uparrow \infty$.
- **NOT tractable** to directly handle exponentially **large** $\mathcal{S}$.

Outline

Scaling up! Then?

What's wrong with current data efficiency solutions?

Introducing HyperAgent: Simple, Efficient, Scalable

- Results

Development of RL Algorithms: A history of "Scale up!"

50s-60s
R. Bellman,
R. A. Howard,
D. Blackwell
Dynamic Programming
(Markov) Decision Processes

70s-90s
R.S. Sutton & A. Barto
C. J.C.H. Watkins
D. Bertsekas & J. Tsitsiklis
Temporal Difference, Q-Learning
Approximate (Neuro) DPs
Parallel and Distributed Computation

2010s --
DeepMind
OpenAI
Etc.
Deep RL (Advances in Algorithms and Computations)
AlphaGo & GPTs (Foundation Models)

Scale up!

Computation First

- **Scale up**↑ : (S1) Larger↑ state space $\mathcal{S}$; (S2) Data $\mathcal{D}$ accumulated↑ .
- **Modern RL Paradigm**: (S1) Function Approximation (Deep Neural Networks); (S2) Continuous adaptation: Incremental optimization with SGD, Experience Replay and/or Target Network.

Key for **Scalablity**: (K1) **Bounded Per-step Computational Complexity**: 'NOT Scale' with $|\mathcal{S}|$ and $|\mathcal{D}|$.

Scalablity $\neq$ Efficiency: Go

Scale up $\uparrow$ AlphaGo$\rightarrow$MuZero Series

[Silver et al., 2016, 2017, 2018, Schrittwieser et al., 2020]

- $\mathcal{S}\uparrow$: Go$\rightarrow$+Board game$\rightarrow$+Atari.
- $\mathcal{D}\uparrow$: human-played games (offline) + self-play (online) $\rightarrow$ Purely self-play (online).

Extremely Inefficient $\downarrow$ (e.g. AlphaGo Zero)

- **Data hungry**: 29 million ($> 10^7$) games of self-play
- **Huge computation costs**: Replication would cost $\approx$ \$35,354,222 due to data collection (sampled from simulated environment) and model computation. Training over 40 days.

Scalablity $\neq$ Efficiency: Standard Atari Benchmarks

Scale up $\uparrow$

- $\mathcal{S}\uparrow$: high-dimensional visual input
- $\mathcal{D}\uparrow$: handle increasingly large amount of game-playing frames

Inefficient $\downarrow$

- **Data**: DQN[Mnih et al., 2015] requires $\approx 200M$ frames to reach human-level performance in Atari.
- **Deployment**: BBF [Schwarzer et al., 2023] combines > 15 heuristics and tricks. Hard and laborious to tune, train and deploy.

Scalablity $\neq$ Efficiency: RLHF for LLMs

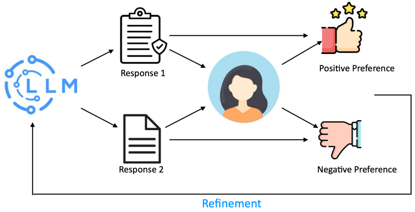


Table 4: E2E time breakdown for training a 13 billion parameter ChatGPT model via DeepSpeed-Chat on a single DGX node with 8 NVIDIA A100-40G GPUs.

Model Sizes	Step 1	Step 2	Step 3	Total
Actor: OPT-13B, Reward: OPT-350M	2.5hr	0.25hr	10.8hr	13.6hr

Table 5: E2E time breakdown for training a 66 billion parameter ChatGPT model via DeepSpeed-Chat on 8 DGX nodes with 8 NVIDIA A100-80G GPUs/node.

Model Sizes	Step 1	Step 2	Step 3	Total
Actor: OPT-66B, Reward: OPT-350M	82 mins	5 mins	7.5hr	9hr

Scale up $\uparrow$

- $\mathcal{S}\uparrow$: more complex, diverse or longer conversations
- $\mathcal{D}\uparrow$: incrementally adapt to extensive online human feedbacks

Inefficient $\downarrow$

- **Data**: **Human feedback** is **scarce** and **expensive** in alignment problem. (1.5M (Offline) and 1.7M (Online) in LLaMA2 [Touvron et al., 2023])
- **Computation**: RLHF occupies most of the training time. [Yao et al., 2023]

Efficiency Challenges in Modern RL: Summary

Modern RL is **scalable**↑, much success in simulated environment.

- **Modern RL Paradigm**: (S1) DNN; (S2) Incremental optimization
- $\Rightarrow$ **Scalable** Algorithm to handle (S1) $\mathcal{S}\uparrow$; and (S2) $\mathcal{D}\uparrow$ with (K1) **Bounded Per-step Computational Complexity**.

Modern RL is **inefficient**↓, an obstacle for **real-world applications**.

- **(E1) Data** Hungry: Collecting data can be **expensive** and **time-consuming** in **real-world**.
- **(E2) Computation**: The per-step computation cost, although bounded (K1), is still high since **Increasingly larger deep network**, e.g. AlphaGo ($> 30M$), GPT-3.5 ($175B$) and GPT-4 ($> 1T$).
- **(E3) Deployment**: **Many heuristic** training tricks and complicated components. Laborious to tune and deploy. Engineering cost is high, especially for **real-world** applications.

Efficiency Challenges in Modern RL: Summary

Modern RL is **scalable**$\uparrow$, much success in simulated environment.

- **Modern RL Paradigm**: (S1) DNN; (S2) Incremental optimization
- $\Rightarrow$ **Scalable** Algorithm to handle (S1) $\mathcal{S}\uparrow$; and (S2) $\mathcal{D}\uparrow$ with (K1) **Bounded Per-step Computational Complexity**.

Modern RL is **inefficient**$\downarrow$, an obstacle for **real-world applications**.

- **(E1) Data** Hungry: Collecting data can be **expensive** and **time-consuming** in **real-world**.
- **(E2) Computation**: The per-step computation cost, although bounded (K1), is still high since **Increasingly larger deep network**, e.g. AlphaGo ($> 30M$), GPT-3.5 ($175B$) and GPT-4 ($> 1T$).
- **(E3) Deployment**: **Many heuristic** training tricks and complicated components. Laborious to tune and deploy. Engineering cost is high, especially for **real-world** applications.

Research question?

Towards fulfilling the promise of RL in real-world complex environment, can we design

- (A1) **Simple** Algorithm — easy to use and deploy (E3)
- (A2) **Efficient** Algorithm — low data (E1) and computation cost (E2)
- (A3) **Scalable** Algorithm — large $\mathcal{S} \uparrow$ (S1) and accumulated $\mathcal{D} \uparrow$ (S2)

"To complicate is easy. To simplify is difficult." – Bruno Munari

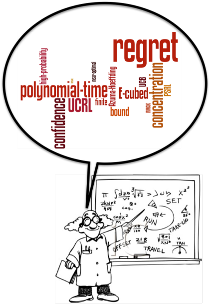
Outline

Scaling up! Then?

What's wrong with current data efficiency solutions?

Introducing HyperAgent: Simple, Efficient, Scalable

Results



R.A. Fisher, William R. Thompson (30s)
T. L. Lai and H. Robbins (80s)
Sequential design and allocations (MAB)
Before 90s

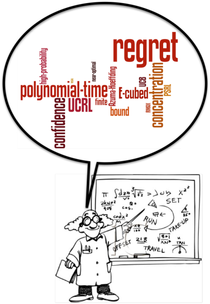
E3/PSRL/UCRL ...
Tabular RL
2000 --

Eluder dimension/Information Ratio
Bellman Rank/Bilinear Structure
Decoupling coefficient/DEC/GEC ...
Structural assumption and Algorithms for RL with function approximation
2014 --

Data First

- ✓ **Provably data efficient exploration strategy**: Optimism in the face of uncertainty (OFU), Posterior sampling, ... (From Tabular era to Function Approximation (FA) era)
- ✓ **Theoretical advancement**: structural assumptions under which RL(FA) is statistically tractable.

Data Efficiency under Function Approximation: Theoretical Effort



R.A. Fisher, William R. Thompson (30s)
T. L. Lai and H. Robbins (80s)
Sequential design and allocations (MAB)
Before 90s

E3/PSRL/UCRL ...
Tabular RL
2000 --

Eluder dimension/Information Ratio
Bellman Rank/Bilinear Structure
Decoupling coefficient/DEC/GEC ...
Structural assumption and Algorithms for RL with function approximation
2014 --

Data First

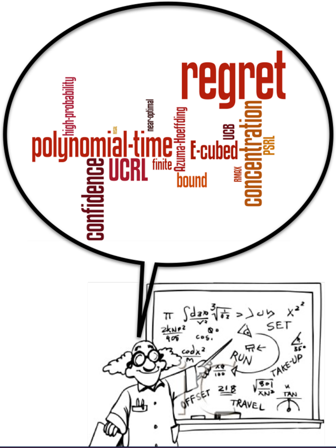
- ✗ **Intractable computation**: intricate nonconvex optimization [Jiang et al., 2017, Jin et al., 2021, Du et al., 2021, Foster et al., 2021, Liu et al., 2023] or sampling from intricate distribution [Zhang, 2022, Dann et al., 2021, Zhong et al., 2022].
- ✗ **Unbounded memory and computation**: e.g. need to re-train entire history for each episode (with regression oracle) [Osband et al., 2019, Wang et al., 2020, Ishfaq et al., 2021, Agarwal et al., 2023]

Data Efficiency in Deep RL: Practical Work

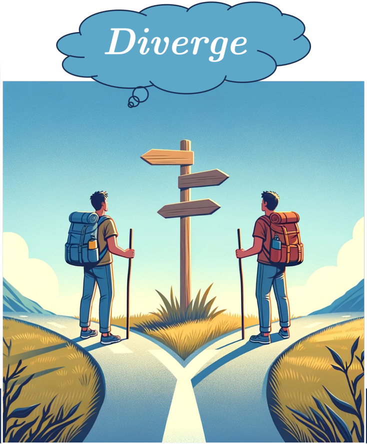
Algorithm	Components
DDQN	incremental SGD with experience replay and target network
Rainbow	(DDQN) + Prioritized replay, Dueling networks, Distributional RL, Noisy Nets.
BBF(23)	(DDQN) + Prioritized replay, Dueling networks, Distributional RL, Self-Prediction, Harder resets, Larger network, Annealing hyper-parameters.

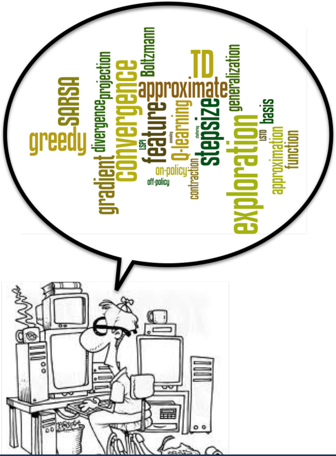
Table: The extra techniques used in different algorithms, e.g. DDQN [Van Hasselt et al., 2016], Rainbow [Hessel et al., 2018], BBF [Schwarzer et al., 2023].

- ✓ **Scalable**: e.g. DDQN use incremental SGD with experience replay and target network.
- ✗ **Not Simple**: Complicated component and many heuristic tricks. Hard and laborious to tune.
- ✗ **Not Efficient**: **Provably inefficient**: e.g. BBF use ϵ-greedy which need exponential many sample in some environment, provably [Kakade, 2003, Strehl, 2007, Osband et al., 2019, Dann et al., 2022].
  Practically inefficient: Per-step computational cost is high, e.g. BBF uses larger networks.

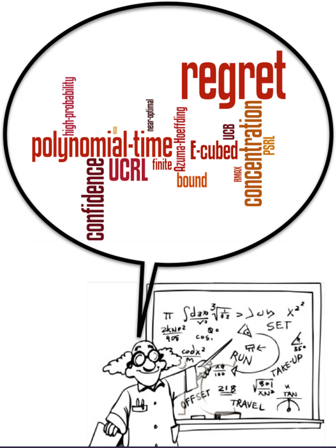


Theory! Data First

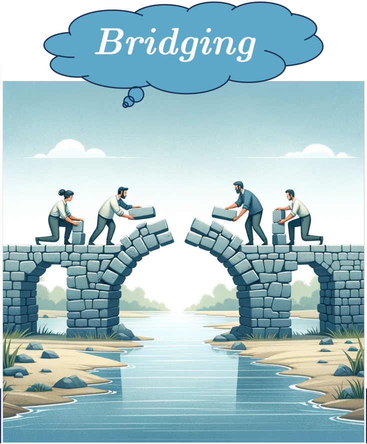


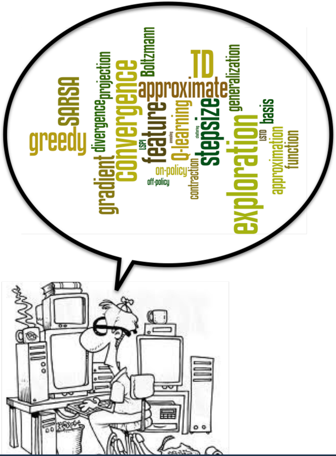


Scale up! Computation First



Theory! Data First





Scale up! Computation First

Outline

Scaling up! Then?

What's wrong with current data efficiency solutions?

Introducing HyperAgent: Simple, Efficient, Scalable

Results

Outline

Scaling up! Then?

What's wrong with current data efficiency solutions?

Introducing HyperAgent: Simple, Efficient, Scalable
    Results

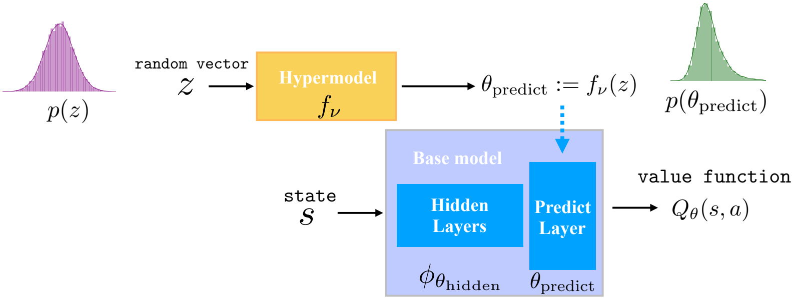
HyperAgent: Simple and Scalable Algorithmic Component

Algorithm	Components
DDQN	incremental SGD with experience replay and target network
Rainbow	(DDQN) + Prioritized replay, Dueling networks, Distributional RL, Noisy Nets.
BBF (23)	(DDQN) + Prioritized replay, Dueling networks, Distributional RL, Self-Prediction, Harder resets, Larger network, Annealing hyper-parameters.
HyperAgent	**Hypermodel**

Table: The extra techniques used in different algorithms, e.g. DDQN [Van Hasselt et al., 2016], Rainbow [Hessel et al., 2018], BBF [Schwarzer et al., 2023] and **our HyperAgent**.

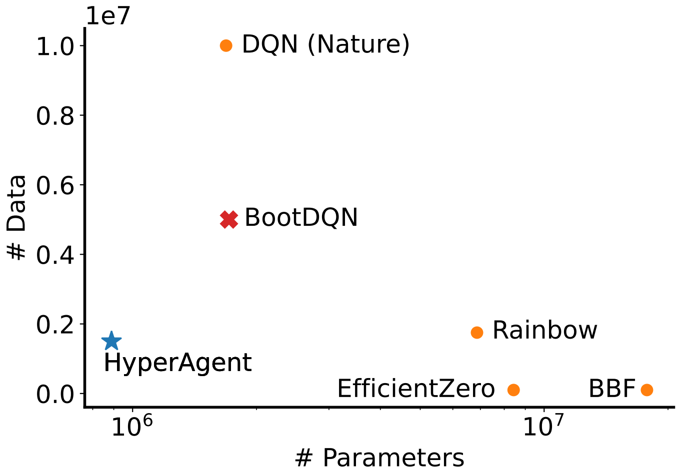
- ✓ **Simple**: Compared to DDQN [Van Hasselt et al., 2016], only one additional component, hypermodel, that is easily compatiable with all Feedforward Deep Networks.
- ✓ **Scalable**: Incremental SGD under DNN function approximation, same as DDQN.

- Base model: DQN-type structure $Q_\theta(s,a) = \langle \phi_{\theta_{\text{hidden}}}(s), \theta_{\text{predict}}(a) \rangle$.
- Hypermodel: $\theta_{\text{predict}} = f_\nu(z)$ where $z \sim p(z)$. $p(z)$ is a fixed reference distribution.



Resulting model: $Q_{\theta_{\text{hidden}}, f_\nu(z)}(s,a)$ is a randomized value function depends on (s,a) and additional random variable z.

HyperAgent: Efficiency in benchmarks (Atari)



- ✓ **Data efficient**: 15% data consumption of DQN[Mnih et al., 2015] by **Deepmind**.
- ✓ **Computation efficient**: 5% model parameters of BBF[Schwarzer et al., 2023] by **Deepmind**.

HyperAgent: Efficiency in benchmarks (DeepSea)

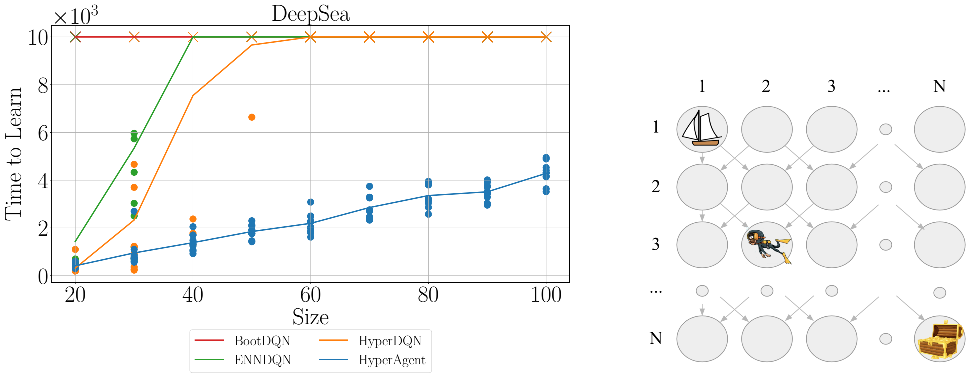


Figure: Comparative results on DeepSea with BootDQN [Osband et al., 2018], HyperDQN [Li et al., 2022], ENN-DQN[Osband et al., 2023]. The y-axis represents the number of episodes required to learn the optimal policy for a specific problem size. The symbol $\times$ indicates the algorithm was unable to learn within 10^4 episodes.

- ✓ scalable as size ↑. ✓ data efficient: optimal episode complexity is linear in the size of the problem.

HyperAgent: Efficiency in 8 hard exploration tasks

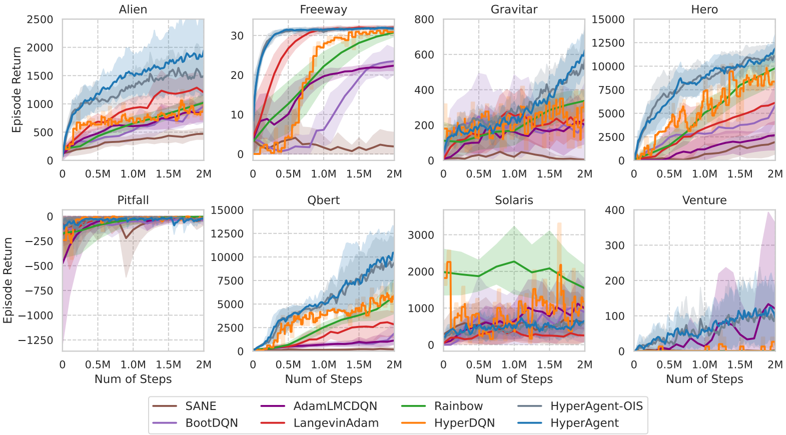


Figure: Comparative results on 8 hardest exploration games. HyperAgent shows stable performance and exploration efficiency compared with randomized RL algorithm including other approximate posterior sampling methods.

HyperAgent: Theoretical Guarantees in RL

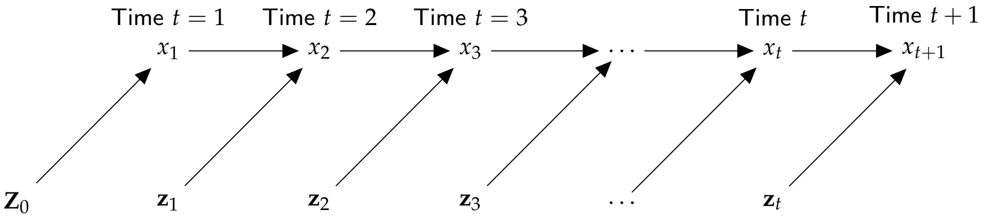
	Practical General FA			Theoreical Finite-horizon Tabular	
Algorithm	FA	Incremental	Efficiency	Regret	Per-step computation
PSRL[Osband and Van Roy, 2017]	✗	✗	✗	$H^2\sqrt{SAK}$	✓ S^2A
RLSVI[Osband et al., 2019]	✓	✗	✗	$H^2\sqrt{SAK}$	✓ S^2A
Ensemble+[Osband et al., 2019]	✓	✓	●	N/A	N/A
Bayes-UCBVI[Tiapkin et al., 2022]	✗	✗	✗	$\sqrt{H^3SAK}$	✓ S^2A
Incre-Bayes-UCBVI[Tiapkin et al., 2022]	✓	✓	●	N/A	N/A
LMC-LSVI[Ishfaq et al., 2023]	✓	✓	●	$H^2\sqrt{S^3A^3K}$	✗ $K \cdot S^2A \cdot \log SAHK$
HyperAgent	✓	✓	✓	$H^2\sqrt{SAK}$	✓ $S^2A \cdot \log SAHK$

- Finite-horizon tabular: # states: S, # actions: A, horizons: H, # episodes: K
- PSRL and Bayes-UCBVI requires dirichlet prior over transitions, otherwise computation intractable; RLSVI requires gaussian noise, otherwise unbounded per-step computation $\tilde{O}(K)$.
- The lemma 3 in [Osband and Van Roy, 2017] target for time-homogeneous MDP may not be correct as pointed out in [Qian et al., 2020]. By a careful revisit, the bound can be corrected to $H^2\sqrt{SAK}$ for time-inhomogeneous setting.

HyperAgent: Possible theoertical extensions

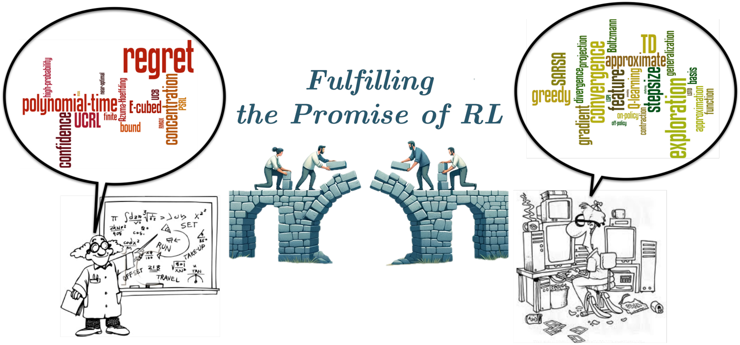
- We already have a theoretical results in linear bandit, which RL with $S = 1, H = 1$ and linear function approximation.
- Immidiate extension to RL under Linear Function Approximation $(H > 1)$ pose no much more difficulty.
- Extension to infinite horizon average-reward RL is doable. I have some preliminary results.
- Extension to function approximation with generalized linear model and neural tangent kernel is possible.

The novelty and difficulty in the mathemtical analysis: No Prior Art



First probability tool for **sequential random projection**. A **Non-trivial** martingale extension of the Johnson–Lindenstrauss lemma and Subspace embedding.

- **Difficulty**: Sequential dependence of high-dimensional R.V. due to the adaptive nature of Sequential Decision Making.
- **Novelty**: A novel and careful construction of stopped process with non-trivial application of 'method of mixtures' in self-normalized martingale.



HyperAgent is the first principled RL agent that is

- **Simple, Efficient** and **Scalable**;
- **Empirically** and **Theoretically** justified. No Prior Art.

References I

A. Agarwal, Y. Jin, and T. Zhang. Vo q l: Towards optimal regret in model-free rl with nonlinear function approximation. In *The Thirty Sixth Annual Conference on Learning Theory*, pages 987–1063. PMLR, 2023.

C. Dann, M. Mohri, T. Zhang, and J. Zimmert. A provably efficient model-free posterior sampling method for episodic reinforcement learning. *Advances in Neural Information Processing Systems*, 34: 12040–12051, 2021.

C. Dann, Y. Mansour, M. Mohri, A. Sekhari, and K. Sridharan. Guarantees for epsilon-greedy reinforcement learning with function approximation. In *International conference on machine learning*, pages 4666–4689. PMLR, 2022.

S. Du, S. Kakade, J. Lee, S. Lovett, G. Mahajan, W. Sun, and R. Wang. Bilinear classes: A structural framework for provable generalization in rl. In *International Conference on Machine Learning*, pages 2826–2836. PMLR, 2021.

D. J. Foster, S. M. Kakade, J. Qian, and A. Rakhlin. The statistical complexity of interactive decision making. *arXiv preprint arXiv:2112.13487*, 2021.

References II

M. Hessel, J. Modayil, H. Van Hasselt, T. Schaul, G. Ostrovski, W. Dabney, D. Horgan, B. Piot, M. Azar, and D. Silver. Rainbow: Combining improvements in deep reinforcement learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

H. Ishfaq, Q. Cui, V. Nguyen, A. Ayoub, Z. Yang, Z. Wang, D. Precup, and L. Yang. Randomized exploration in reinforcement learning with general value function approximation. In *International Conference on Machine Learning*, pages 4607–4616. PMLR, 2021.

H. Ishfaq, Q. Lan, P. Xu, A. R. Mahmood, D. Precup, A. Anandkumar, and K. Azizzadenesheli. Provable and practical: Efficient exploration in reinforcement learning via langevin monte carlo, 2023.

N. Jiang, A. Krishnamurthy, A. Agarwal, J. Langford, and R. E. Schapire. Contextual decision processes with low bellman rank are pac-learnable. In *International Conference on Machine Learning*, pages 1704–1713. PMLR, 2017.

C. Jin, Q. Liu, and S. Miryoosefi. Bellman eluder dimension: New rich classes of rl problems, and sample-efficient algorithms. *Advances in neural information processing systems*, 34:13406–13418, 2021.

References III

S. M. Kakade. *On the sample complexity of reinforcement learning*. University of London, University College London (United Kingdom), 2003.

Z. Li, Y. Li, Y. Zhang, T. Zhang, and Z.-Q. Luo. HyperDQN: A randomized exploration method for deep reinforcement learning. In *International Conference on Learning Representations*, 2022. URL `https://openreview.net/forum?id=X0nrKAXu7g-`.

Z. Liu, M. Lu, W. Xiong, H. Zhong, H. Hu, S. Zhang, S. Zheng, Z. Yang, and Z. Wang. Maximize to explore: One objective function fusing estimation, planning, and exploration. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL `https://openreview.net/forum?id=A57UMlUJdc`.

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.

References IV

I. Osband and B. Van Roy. Why is posterior sampling better than optimism for reinforcement learning? In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, page 2701–2710. JMLR.org, 2017.

I. Osband, J. Aslanides, and A. Cassirer. Randomized prior functions for deep reinforcement learning. *Advances in Neural Information Processing Systems*, 31, 2018.

I. Osband, B. V. Roy, D. J. Russo, and Z. Wen. Deep exploration via randomized value functions. *Journal of Machine Learning Research*, 20(124):1–62, 2019. URL `http://jmlr.org/papers/v20/18-339.html`.

I. Osband, Z. Wen, S. M. Asghari, V. Dwaracherla, M. Ibrahimi, X. Lu, and B. Van Roy. Approximate thompson sampling via epistemic neural networks. *arXiv preprint arXiv:2302.09205*, 2023.

J. Qian, R. Fruit, M. Pirotta, and A. Lazaric. Concentration inequalities for multinoulli random variables. *arXiv preprint arXiv:2001.11595*, 2020.

J. Schrittwieser, I. Antonoglou, T. Hubert, K. Simonyan, L. Sifre, S. Schmitt, A. Guez, E. Lockhart, D. Hassabis, T. Graepel, et al. Mastering atari, go, chess and shogi by planning with a learned model. *Nature*, 588(7839):604–609, 2020.

M. Schwarzer, J. S. O. Ceron, A. Courville, M. G. Bellemare, R. Agarwal, and P. S. Castro. Bigger, better, faster: Human-level atari with human-level efficiency. In *International Conference on Machine Learning*, pages 30365–30380. PMLR, 2023.

D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. Van Den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, et al. Mastering the game of go with deep neural networks and tree search. *nature*, 529(7587):484–489, 2016.

D. Silver, J. Schrittwieser, K. Simonyan, I. Antonoglou, A. Huang, A. Guez, T. Hubert, L. Baker, M. Lai, A. Bolton, et al. Mastering the game of go without human knowledge. *nature*, 550(7676):354–359, 2017.

D. Silver, T. Hubert, J. Schrittwieser, I. Antonoglou, M. Lai, A. Guez, M. Lanctot, L. Sifre, D. Kumaran, T. Graepel, et al. A general reinforcement learning algorithm that masters chess, shogi, and go through self-play. *Science*, 362(6419):1140–1144, 2018.

A. L. Strehl. *Probably approximately correct (PAC) exploration in reinforcement learning*. PhD thesis, Rutgers University-Graduate School-New Brunswick, 2007.

D. Tiapkin, D. Belomestny, É. Moulines, A. Naumov, S. Samsonov, Y. Tang, M. Valko, and P. Ménard. From dirichlet to rubin: Optimistic exploration in rl without bonuses. In *International Conference on Machine Learning*, pages 21380–21431. PMLR, 2022.

H. Touvron, L. Martin, K. Stone, P. Albert, A. Almahairi, Y. Babaei, N. Bashlykov, S. Batra, P. Bhargava, S. Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

H. Van Hasselt, A. Guez, and D. Silver. Deep reinforcement learning with double q-learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 30, 2016.

References VII

R. Wang, R. R. Salakhutdinov, and L. Yang. Reinforcement learning with general value function approximation: Provably efficient approach via bounded eluder dimension. *Advances in Neural Information Processing Systems*, 33:6123–6135, 2020.

Z. Yao, R. Y. Aminabadi, O. Ruwase, S. Rajbhandari, X. Wu, A. A. Awan, J. Rasley, M. Zhang, C. Li, C. Holmes, et al. Deepspeed-chat: Easy, fast and affordable rlhf training of chatgpt-like models at all scales. *arXiv preprint arXiv:2308.01320*, 2023.

T. Zhang. Feel-good thompson sampling for contextual bandits and reinforcement learning. *SIAM Journal on Mathematics of Data Science*, 4(2):834–857, 2022.

H. Zhong, W. Xiong, S. Zheng, L. Wang, Z. Wang, Z. Yang, and T. Zhang. Gec: A unified framework for interactive decision making in mdp, pomdp, and beyond. *arXiv preprint arXiv:2211.01962*, 2022.