模式分析与机器智能
工业和信息化部重点实验室
MIIT Key Laboratory of
Pattern Analysis & Machine Intelligence

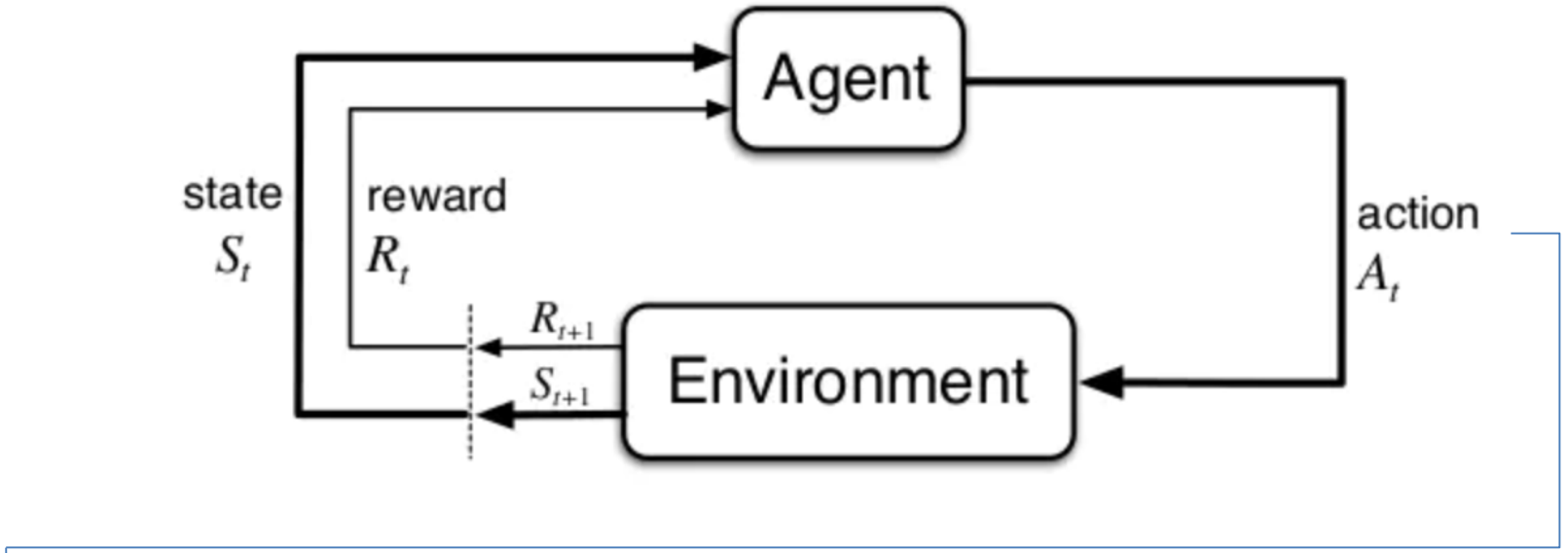
ParNeC 模式识别与神经计算研究组
PAttern Recognition and NEural Computing

HyperDQN: A Randomized Exploration Method for Deep Reinforcement Learning

Ziniu Li[1], Yingru Li[1†], Yushun Zhang[1], Tong Zhang[2†], and Zhi-Quan Luo[1†]
[1]Shenzhen Research Institute of Big Data,
The Chinese University of Hong Kong, Shenzhen, China
[2]Hong Kong University of Science and Technology
{ziniuli, yingruli, yushunzhang}@link.cuhk.edu.cn,
tongzhang@ust.hk, luozq@cuhk.edu.cn

ICLR 2022

Problem: Exploration and Exploitation



Exploitation ⇒ higher immediate reward & local optimum

Exploration ⇒ sacrifice immediate reward & wider space

Classic exploration methods

1.ε-greedy : Select greedy action with probability of (1-ε)
Select action randomly with probability of ε
ε decreases with the increase of the agent step

2.Boltzmann distribution : Select action according to the Boltzmann distribution of Q-value (softmax)

3.Upper confidence bounds(UCB) : $A_t{}^{UCB} \doteq arg\max_a \left[Q_t + c\sqrt{\frac{\ln t}{N_t(a)}} \right]$

$N_t(a)$: The number of aciton a selected before time t

4.Thomson Sample : posterior sampling

(1)Set up Q-value distribution estimates for each action
(2)Sample randomly from each distribution to get Q-value,and select the action with the maximum Q-value
(3)Update distribution parameters based on reward

Benefits of posterior sampling

- The upper regret bound is reduced

$$\mathrm{Regret}(T, \mathcal{M}) = \sum_{l=0}^{T/H-1} \mathbb{E}_{\mathcal{M}} \left[V_0^*(s_{l0}) - \sum_{h=0}^{H} r_{lh} \right]$$

- The randomness in posterior sampling could yield positive bias, which boosts optimistic behaviors

- the property of temporal extended exploration(deep exploration)

Randomized least-squares value iteration（RLSVI)

Algorithm 2 RLSVI with greedy action

Input: Features $\Phi_0,..,\Phi_{H-1}$; $\sigma>0$, $\lambda>0$

1: **for** $l=0,1,..$ **do**
2: Compute $\tilde{\theta}_{l0},..,\tilde{\theta}_{l,H-1}$ using Algorithm 1
3: Observe s_{l0}
4: **for** $h=0,..,H-1$ **do**
5: Sample $a_{lh}\in\text{argmax}_{\alpha\in\mathcal{A}}\left(\Phi_h\tilde{\theta}_{lh}\right)(s_{lh},\alpha)$
6: Observe r_{lh} and $s_{l,h+1}$
7: **end for**
8: Observe r_{lH}
9: **end for**

Step1: Given initial value of θ (l=0)
Step2: Q-value comes from Φθ,and select action with greedy strategy
Step3: Update θ according to Bayesian regression
Step4: Repeat steps 2 and 3 for each episode

Algorithm 1 Randomized Least-Squares Value Iteration

Input: Data $\Phi_0(s_{i0},a_{i0}),r_{i0},..,\Phi_{H-1}(s_{iH-1},a_{iH-1}),r_{iH}$: $i<L$, Parameters $\lambda>0$, $\sigma>0$

Output: $\tilde{\theta}_{l0},..,\tilde{\theta}_{l,H-1}$

1: **for** $h=H-1,..,1,0$ **do**
2: Generate regression problem $A\in\mathbb{R}^{l\times K}$, $b\in\mathbb{R}^l$:

$$A\leftarrow\begin{bmatrix}\Phi_h(s_{0h},a_{0h})\\ \vdots\\ \Phi_h(s_{l-1,h},a_{l-1,h})\end{bmatrix}$$

$$b_i\leftarrow\begin{cases}r_{ih}+\max_\alpha\left(\Phi_{h+1}\tilde{\theta}_{l,h+1}\right)(s_{i,h+1},\alpha) & \text{if } h<H-1\\ r_{ih}+r_{i,h+1} & \text{if } h=H-1\end{cases}$$

3: Bayesian linear regression for the value function

$$\bar{\theta}_{lh}\leftarrow\frac{1}{\sigma^2}\left(\frac{1}{\sigma^2}A^\top A+\lambda I\right)^{-1}A^\top b$$

$$\Sigma_{lh}\leftarrow\left(\frac{1}{\sigma^2}A^\top A+\lambda I\right)^{-1}$$

4: Sample $\tilde{\theta}_{lh}\sim N(\bar{\theta}_{lh},\Sigma_{lh})$ from Gaussian posterior
5: **end for**

Bayesian Linear Regression

conditions: $y = \langle \theta^*, x \rangle + \omega^*,\ \omega^* \sim N(0, \sigma_\omega^2)$ prior distribution $p_0(\theta^*) \sim N(\bar{\theta}_p, \sigma_p^2)$ Dataset $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^N$

According to Bayes formula:

$$p(\theta^*|\mathcal{D}) = \frac{p(\mathcal{D}|\theta^*)p_0(\theta^*)}{p(\mathcal{D})} \Rightarrow p(\theta^*|\mathcal{D}) \propto p(\mathcal{D}|\theta^*)p_0(\theta^*) \Rightarrow p(\mathcal{D}|\theta^*) = p(Y|\theta^*, X) \sim N(X\theta^*, \sigma_\omega^2)$$

Conjugate prior: For some likelihood functions, its prior and posterior have the same distribution

Multi − dimensional gaussian distribution:

$$f(x) = \frac{1}{(2\pi)^{\frac{n}{2}}|\Sigma|^{\frac{1}{2}}} \exp\left\{-\frac{1}{2}(X-\mu)^T\Sigma^{-1}(X-\mu)\right\}, \quad \Sigma\text{: Covariance matrix}$$

So we can assume that the distribution: $p(\theta^*|\mathcal{D}) \sim N(\mu, \Sigma)$

For an ordinary multidimensional Gaussian distribution:

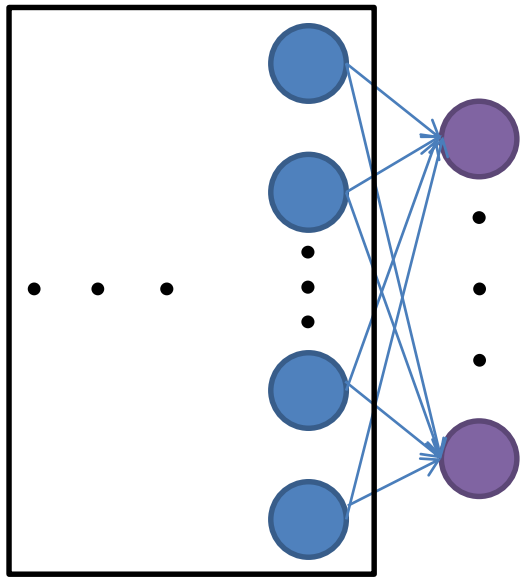
$$-\frac{1}{2}(X-\mu)^T\Sigma^{-1}(X-\mu) = -\frac{1}{2}\left(X^T\Sigma^{-1}X - X^T\Sigma^{-1}\mu - \mu^T\Sigma^{-1}X + \mu^T\Sigma^{-1}\mu\right)$$

$$\begin{aligned}
p(\mathcal{D}|\theta^*) \propto p(\mathcal{D}|\theta^*)p_0(\theta^*) &\propto \exp\left\{-\frac{1}{2\sigma_\omega^2}(Y^T - \theta^{*T}X^T)(Y - X\theta^*)\right\}\exp\left\{-\frac{1}{2\sigma_p^2}(\theta^* - \bar{\theta}_p)^T(\theta^* - \bar{\theta}_p)\right\} \\
&= \exp\left\{-\frac{1}{2\sigma_\omega^2}(Y^TY - \theta^{*T}X^TY - Y^TX\theta^* + \theta^{*T}X^TX\theta^*) - \frac{1}{2\sigma_p^2}(\theta^* - \bar{\theta}_p)^T(\theta^* - \bar{\theta}_p)\right\} \\
&= \exp\left\{-\frac{1}{2\sigma_\omega^2}(Y^TY - \theta^{*T}X^TY - Y^TX\theta^* + \theta^{*T}X^TX\theta^*) - \frac{1}{2}(\theta^* - \bar{\theta}_p)^T(\theta^* - \bar{\theta}_p)\right\} \\
&= \exp\left\{-\frac{1}{2}\left[\theta^{*T}(\sigma_\omega^{-2}X^TX + \sigma_p^{-2}I)\theta^*\right] + \frac{1}{2}\left[\theta^{*T}(\sigma_\omega^{-2}X^TY + \sigma_p^{-2}\bar{\theta}_p) + (Y^TX\sigma_\omega^{-2} + \bar{\theta}_p{}^T\sigma_p^{-2})\theta^*\right] + c\right\}
\end{aligned}$$

Therefore $\Sigma = \sigma_\omega^{-2}X^TX + \sigma_p^{-2}I \quad \mu = (\sigma_\omega^{-2}X^TX + \sigma_p^{-2}I)(\sigma_\omega^{-2}X^TY + \sigma_p^{-2}\bar{\theta}_p)$

Problem in extending to DRL

- Provided feature Φ



Features Φ Q

Algorithm 2 RLSVI with greedy action

Input: Features $\Phi_0,..,\Phi_{H-1}$; $\sigma>0$, $\lambda>0$

1: **for** $l=0,1,..$ **do**
2: Compute $\tilde{\theta}_{l0},..,\tilde{\theta}_{l,H-1}$ using Algorithm 1
3: Observe s_{l0}
4: **for** $h=0,..,H-1$ **do**
5: Sample $a_{lh}\in\text{argmax}_{\alpha\in\mathcal{A}}\left(\Phi_h\tilde{\theta}_{lh}\right)(s_{lh},\alpha)$
6: Observe r_{lh} and $s_{l,h+1}$
7: **end for**
8: Observe r_{lH}
9: **end for**

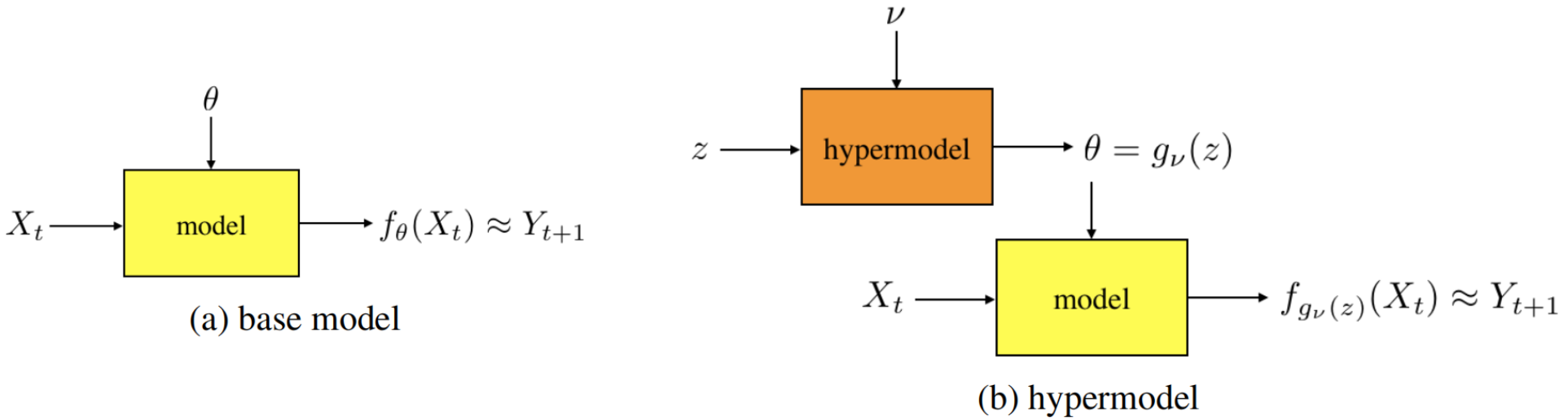
- The computational complexity

$$\text{fixed } \phi: \quad \Phi_K = \Phi_{K-1} + \phi(x_K)\phi(x_K)^\top \quad \text{with } \Phi_0 = \mathbf{0},$$

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

Covariance matrix

Hypermodel



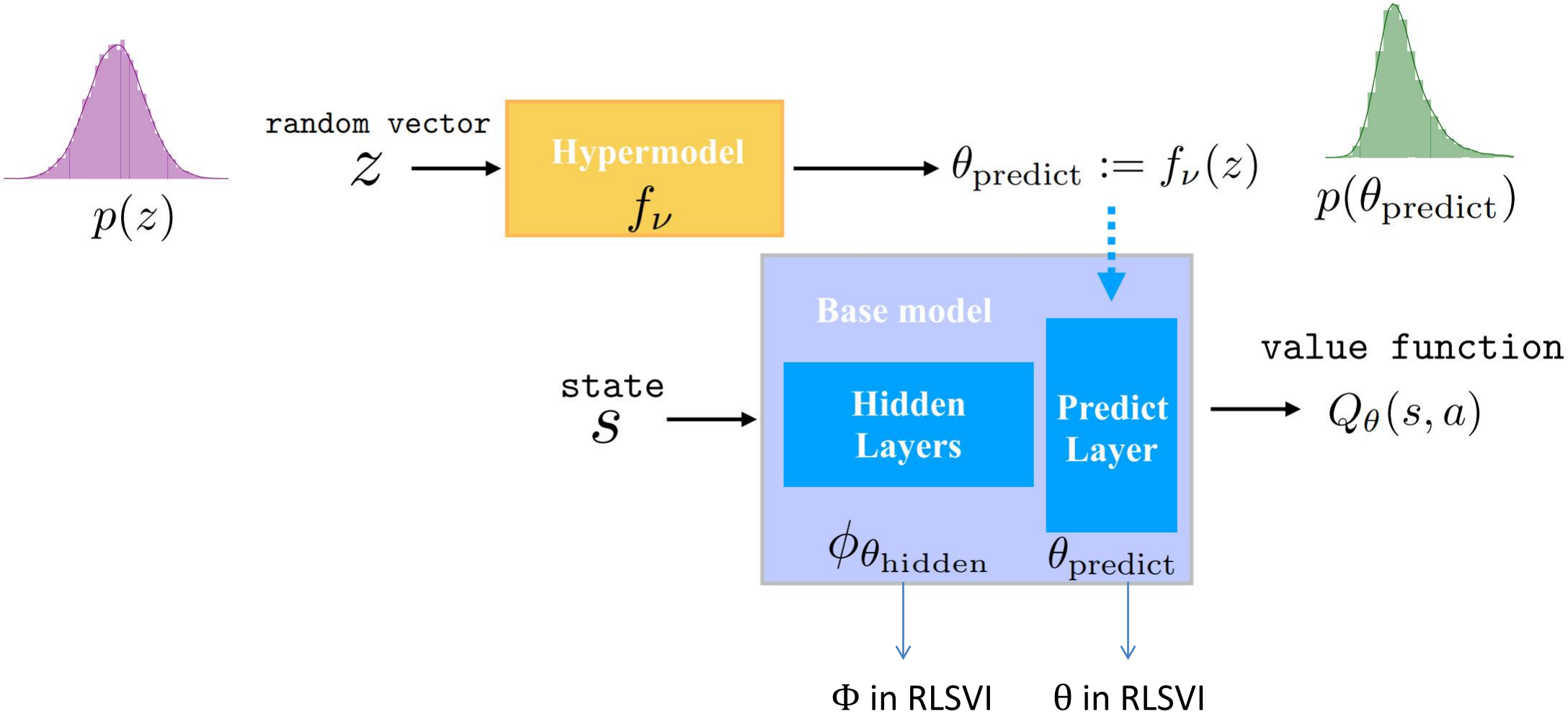
Theorem 1 *Let p_z be the unit Gaussian distribution in $\Re^K$. For all $\epsilon > 0$, $\delta > 0$, $B > 0$, and probability measures μ over $L_\infty(\mathcal{X}, B)$, there exists a transport map H from p_z to μ, a neural network $f_\theta : \mathcal{X} \mapsto \Re$ with a linear output node and ReLU hidden nodes, and a linear hypermodel $g_\nu : \mathcal{Z} \mapsto \Re^{N_\theta}$ such that*

$$\|f_{g_\nu(z)} - f^*\|_\infty \leq \epsilon,$$

with probability at least $1 - \delta$, for some $\nu \in \Re^{N_\nu}$, where $f^* = H(z)$.

A linear hypermodel can represent essentially any distribution over functions

HyperDQN


$p(z)$
random vector
z
Hypermodel
f_ν
$\theta_{\text{predict}} := f_\nu(z)$
$p(\theta_{\text{predict}})$
Base model
Hidden Layers
Predict Layer
$\phi\theta_{\text{hidden}}$
θ_{predict}
state
s
value function
$Q_\theta(s,a)$
Φ in RLSVI
θ in RLSVI

Loss function

$$L(\nu;\mathcal{D}) = \int_z p(z)\Bigg[\sum_{(x,y,\xi)\in\mathcal{D}} \Big(y + \underbrace{\sigma_\omega z^\top \xi}_{(a)} - \underbrace{\big(g_{f_{\nu_{\text{prior}}}(z)}(x) + g_{f_\nu(z)}(x)\big)}_{(b)}\Big)^2 + \underbrace{\frac{\sigma_\omega^2}{\sigma_p^2}\left\|f_\nu(z)\right\|^2}_{(c)}\Bigg](\mathrm{d}z), \tag{4.1}$$

$$\min_{\nu,\theta_{\text{hidden}}} \int_z p(z)\Bigg[\sum_{(s,a,r,\xi,s')\in\mathcal{D}} \Big(Q_{\text{target}}(s',z) + \sigma_\omega z^\top \xi - Q_{\text{prediction}}(s,a,z)\Big)^2 + \frac{\sigma_\omega^2}{\sigma_p^2}\left\|f_\nu(z)\right\|^2\Bigg](\mathrm{d}z), \tag{4.2}$$

where

$$\begin{aligned} Q_{\text{prediction}}(s,a,z) &= Q_{\theta_{\text{prior}},f_{\nu_{\text{prior}}}(z)}(s,a) + Q_{\theta_{\text{hidden}},f_\nu(z)}(s,a), \\ Q_{\text{target}}(s',z) &= r + \gamma \max_{a'}\left[Q_{\theta_{\text{prior}},f_{\nu_{\text{prior}}}(z)}(s',a') + Q_{\bar{\theta}_{\text{hidden}},f_{\bar{\nu}}(z)}(s',a')\right]. \end{aligned} \tag{4.3}$$

$p(z)$:Gaussian distribution

ξ:a random vector independently sampled from the unit hypersphere

- Generate $\mathbf{x} \sim N(0, I)$
- Normalize $\mathbf{x}$

$\sigma_\omega z^T \xi$:is an artificial noise term exerted on the label y

$\frac{\sigma_\omega^2}{\sigma_p^2}\left\|f_v(z)\right\|^2$:a regularization term

Theorem

$$L(\nu;\mathcal{D}) = \int_z p(z)\Bigg[\sum_{(x,y,\xi)\in\mathcal{D}} \Big(y + \underbrace{\sigma_\omega z^\top \xi}_{(a)} - \underbrace{\big(g_{f_{\nu_{\text{prior}}}(z)}(x) + g_{f_\nu(z)}(x)\big)}_{(b)}\Big)^2 + \underbrace{\frac{\sigma_\omega^2}{\sigma_p^2}\|f_\nu(z)\|^2}_{(c)}\Bigg](\mathrm{d}z), \tag{4.1}$$

Assumption 1. *Suppose the data generation follows* $y = x^\top\theta^\star + \omega^\star$, $\omega^\star \sim \mathcal{N}(0,\sigma_\omega^2)$ *and the prior distribution over* $\theta^\star$ *is* $\mathcal{N}(\bar{\theta}_p, \sigma_p^2 I)$. *Furthermore, assume the base model is linear, i.e.,* $g_{f_\nu(z)}(x) = x^\top f_\nu(z)$ *and* $g_{f_{\nu_{prior}}(z)}(x) = x^\top f_{\nu_{prior}}(z)$. *Moreover, assume the hypermodel is also linear, i.e.,* $f_\nu(z) = \nu_w^\top z + \nu_b$ *and* $f_{\nu_{prior}}(z) = \nu_w^{prior\top} z + \nu_b^{prior}$; $\nu = (\nu_w, \nu_b)$ *and* $\nu_{prior} = (\nu_w^{prior}, \nu_b^{prior})$.

Theorem 1 (Formal statement)**.** *Under Assumption 1, set* $\nu_w^{prior} = \sigma_p I$ *and* $\nu_b^{prior} = \bar{\theta}_p$. *Let* $\nu^\star = (\nu_w^\star, \nu_b^\star)$ *be the optimal solution of (4.1) conditioned on specific realizations of* ξ, *then* $\theta := f_{\nu_{prior}}(z) + f_{\nu^\star}(z) \sim \mathcal{N}(\nu_b^{prior} + \nu_b^\star, (\nu_w^{prior} + \nu_w^\star)^\top(\nu_w^{prior} + \nu_w^\star))$ *with*

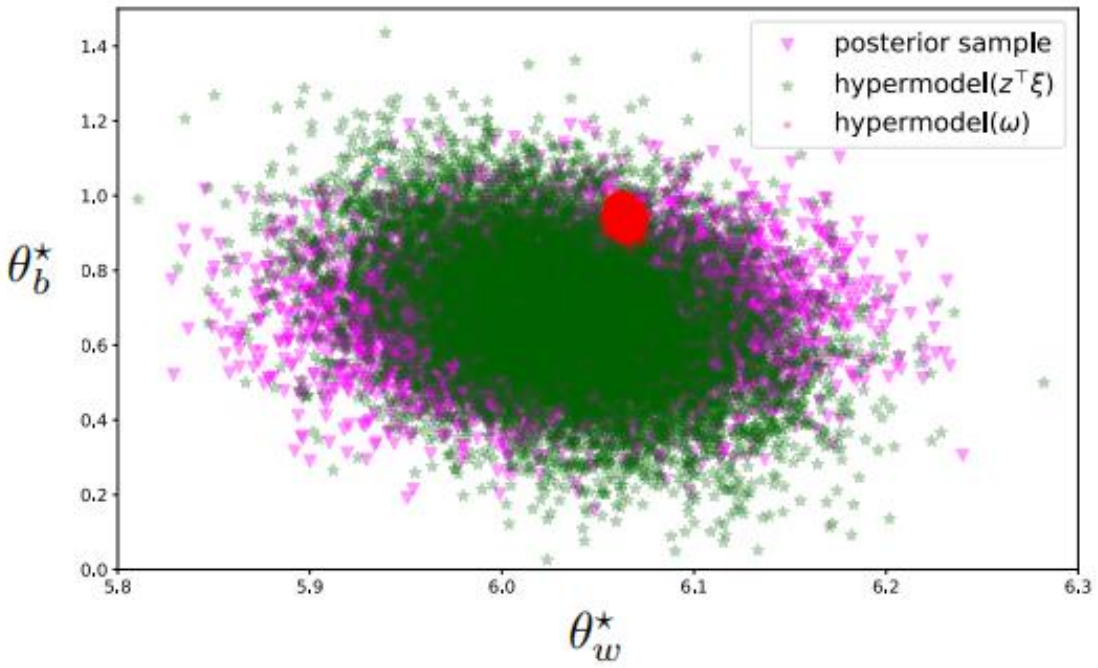
$$\nu_b^{prior} + \nu_b^\star = \mathbb{E}[\theta^\star \mid \mathcal{D}], \quad (\nu_w^{prior} + \nu_w^\star)^\top(\nu_w^{prior} + \nu_w^\star) = \mathrm{Cov}[\theta^\star \mid \mathcal{D}] + \mathtt{err}(\Xi),$$

where

$$\mathtt{err}(\Xi) := \mathrm{Cov}[\theta^\star \mid \mathcal{D}]\left(\frac{1}{\sigma_\omega^2}\sum_{(x,\xi)\neq(x',\xi')\in\mathcal{D}} x\xi^\top\xi' x'^\top + \frac{1}{\sigma_\omega\sigma_p}\sum_{(x,\xi)\in\mathcal{D}}(x\xi^\top + \xi x^\top)\right)\mathrm{Cov}[\theta^\star \mid \mathcal{D}]$$

Furthermore, the error term satisfies $\mathbb{E}_\Xi[\mathtt{err}(\Xi)] = 0$.

Noise choice



z-dependent noise is indispensable

Algorithm

Algorithm 2 HyperDQN

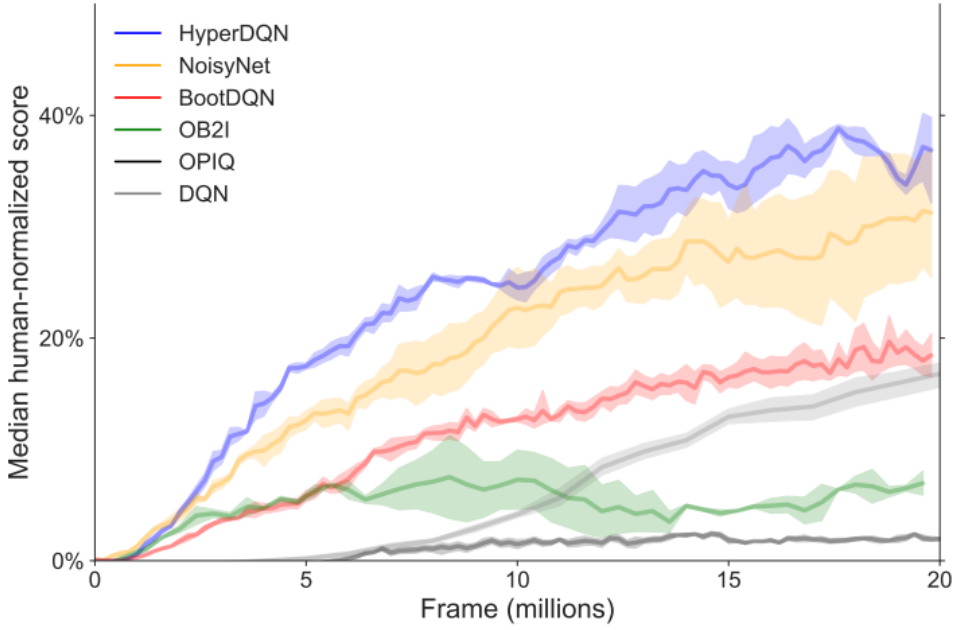
agent step $n \leftarrow 0$, train step $\ell \leftarrow 0$.
for episode $k = 0, 1, 2, \cdots$ **do**
  generate a random vector $z \sim \mathcal{N}(0, I)$. ▷ Sampling
  instantiate the Q-value function $Q_\theta(s, a, z)$.
  for stage $t = 0, 1, 2, \cdots, T-1$ **do** ▷ Interaction
    observe state s_t.
    take the greedy action $a \leftarrow \text{argmax}_a\, Q_\theta(s_t, a, z)$.
    receive the next state s_{t+1} and reward $r(s_t, a_t)$.
    sample ξ uniformly from unit hypersphere.
    store $(s_t, a_t, r, \xi, s_{t+1})$ into the replay buffer $\mathcal{D}$.
    agent step $n \leftarrow n + 1$.
    if mod (agent step n, train frequency M) $== 0$ **then** ▷ Update
      sample a mini-batch $\widetilde{\mathcal{D}}$ of (s, a, r, ξ, s') from the replay buffer $\mathcal{D}$.
      sample N random vectors z_i': $\widetilde{\mathcal{Z}} = \{z_i'\}_{i=1}^N$.
      optimize ν and θ_{hidden} using the empirical loss function (A.6) with $\widetilde{\mathcal{D}}$ and $\widetilde{\mathcal{Z}}$.
      train step $\ell \leftarrow \ell + 1$.
    end if
    if mod (train step ℓ, target update frequency G) $== 0$ **then**
      update the target network.
    end if
  end for
end for

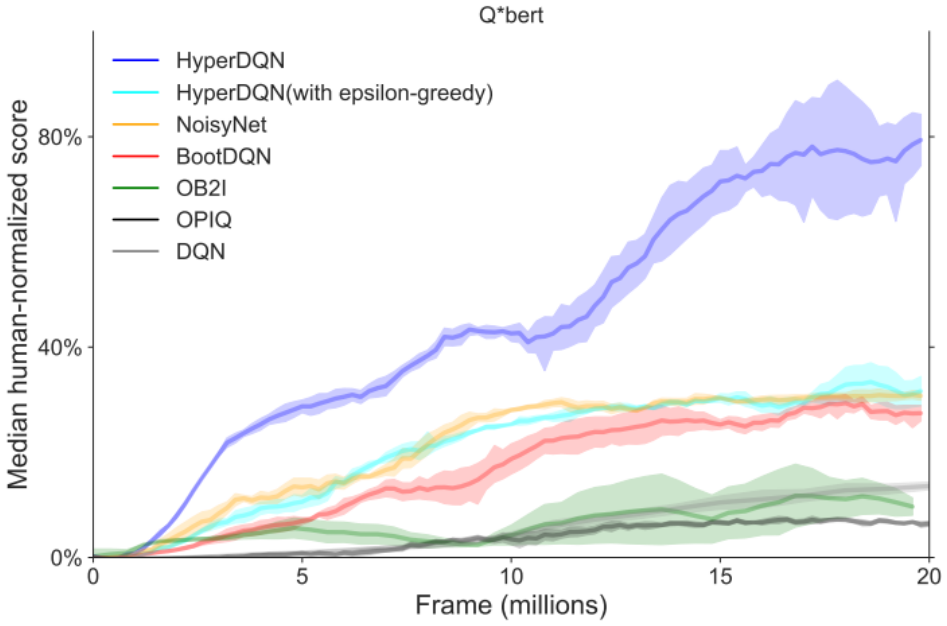
$$\frac{1}{|\widetilde{\mathcal{Z}}|}\left(\sum_{z\in\widetilde{\mathcal{Z}}}\frac{|\mathcal{D}|}{|\widetilde{\mathcal{D}}|}\sum_{(s,a,r,\xi,s')\in\widetilde{\mathcal{D}}}\left(Q_{\text{target}}(s',z)+\sigma_\omega z^\top\xi-Q_{\text{prediction}}(s,a,z)\right)^2+\frac{\sigma_\omega^2}{\sigma_p^2}\left\|f_\nu(z)\right\|^2\right), \tag{A.6}$$

Atari

Table 2: Comparison of algorithms on Atari in terms of the median over 49 games' maximum human-normalized scores. Note that the performance of DQN is based on 200M training frames while other methods are based on 20M training frames.

DQN (**200M**)	OPIQ	OB2I	BootDQN	NoisyNet	HyperDQN
93%	37%	50%	82%	91%	**110%**





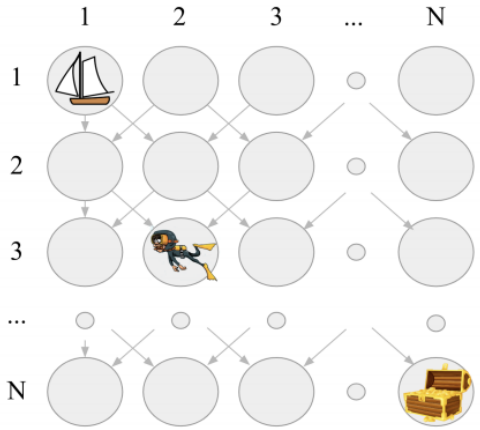
Experiment

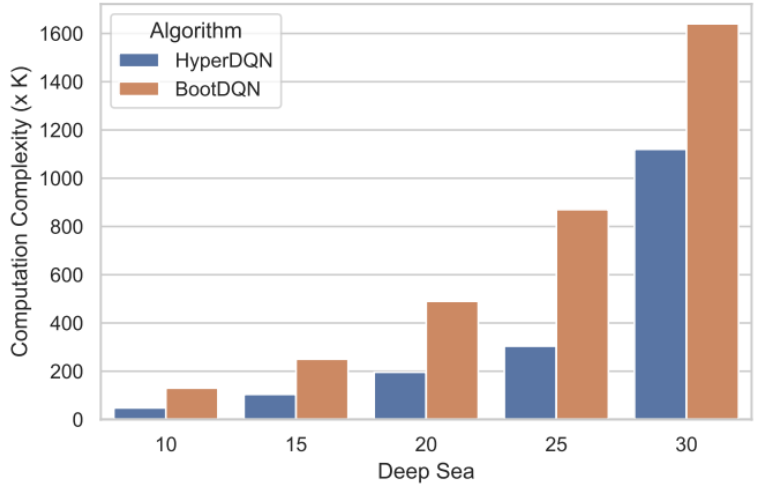
SuperMarioBros

Table 3: Comparison of algorithms on SuperMarioBros in terms of the raw scores by the best policies with 20M training frames.

	DQN	OPIQ	OB2I	BootDQN	NoisyNet	HyperDQN
SuperMarioBros-1-1	1,070	7,650	4,457	7,009	**12,439**	7,924
SuperMarioBros-1-2	2,883	5,515	4,695	5,665	6,347	**8,267**
SuperMarioBros-1-3	667	2,053	1,583	1,609	1,587	**6,047**
SuperMarioBros-2-1	10,800	21,654	14,226	**26,415**	14,017	23,047
SuperMarioBros-2-2	813	1,630	1,588	1,092	1,808	**1,984**
SuperMarioBros-2-3	3,373	4,718	4,402	5,108	**6,490**	5,980
SuperMarioBros-3-1	2,560	3,700	3,251	3,862	11,310	**48,385**
SuperMarioBros-3-2	11,633	20,872	26,508	20,955	33,489	**41,140**
SuperMarioBros-3-3	1,007	2,440	3,009	2,650	**5,886**	5,568

Deep Sea





Thanks