Augmenting Decision with Hypothesis in Reinforcement Learning

Nguyen Minh Quang [1] **Hady W. Lauw** [1]

Abstract

Value-based reinforcement learning is the current State-Of-The-Art due to high sampling efficiency. However, our theoretical and empirical studies show evidence that it suffers from low exploitation in early training period and bias sensitiveness. To address these issues, we propose to augment the decision-making process with hypothesis, a weak form of environment description. Our approach relies on prompting the learning agent with accurate hypotheses, and designing a ready-to-adapt policy through incremental learning. We propose the ALH algorithm, showing detailed analyses on a typical learning scheme and a diverse set of Mujoco benchmarks. Our algorithm produces a significant improvement over value-based learning algorithms and other strong baselines. Our code is available at https://github.com/nbtpj/ALH.

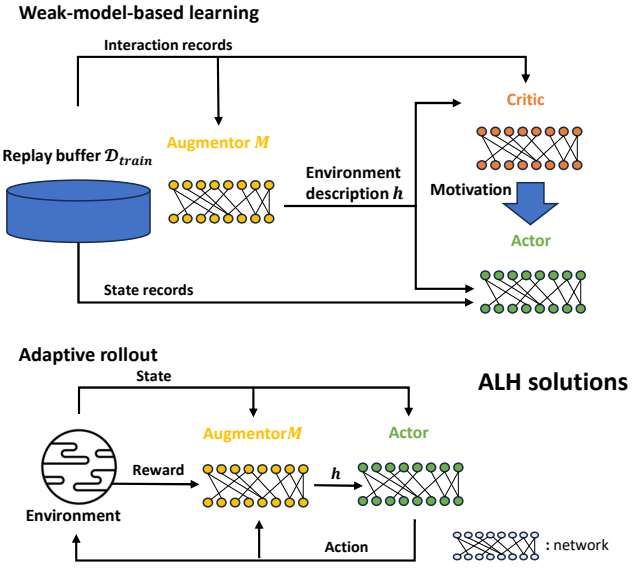

Figure 1. We introduce to augment learners with hypothesis representation, and propose two key hypothesis-based solutions: weak-model-based learning and employing adaptation in rollout.

1. Introduction

Reinforcement learning (RL) is a branch of artificial intelligence that studies how agents can learn to make optimal decisions in complex and dynamic environments, based on their own actions and environment return signal (Sutton & Barto, 2018). In this study, we consider RL in a deterministic Markov Decision Process (MDP) defined by $(\mathcal{S}, \mathcal{A}, \mathcal{T}, \mathcal{R}, \gamma)$ where $\mathcal{S}$ is the state space, $\mathcal{A}$ is the action space, and $\gamma \in [0, 1)$ is the discount factor. The environment is characterized by 2 attributions: transition function $\mathcal{T} : \mathcal{S} \times \mathcal{A} \to \mathcal{S}$; and reward function $\mathcal{R} : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$. The output of reinforcement learning is a policy $\pi : \mathcal{S} \to \mathcal{A}$, which maximizes RL objective $J(\pi) := \mathbb{E}[\sum_{t=0}^{\infty} \gamma^t r_t | a_t = \pi(s_t)]$, where $\mathbb{E}$ denotes the expectation.

Among RL approaches, value-based RL shows higher sampling efficiency in comparison with policy-based optimization (e.g., Schulman et al. (2017)), and lower computational complexity than model-based RL (e.g. Janner et al. (2019)). Value-based RL refers to RL algorithms which mainly rely on future advantage estimation to optimize the policy. A typical value-based learning can be defined by the combination of two optimization processes in Q-learning, whose Q value is the motivation for a action given a certain state:

$$\theta^* = \arg\min_{\theta} \mathbb{E}_{s,a,s',r,d} ||Q_\theta(s,a) - Q_{target}(s,a)||_2 \quad (1)$$

$$\omega^* = \arg\max_{\omega} \mathbb{E}_s Q_\theta(s, \pi_\omega(s)) \quad (2)$$

where $Q_{target}(s,a) = \mathcal{R}(s,a) + \gamma(1-d)\max_{a'} Q(s',a')$ is the Bellman optimal Q value, s, a, s', r, d are respectively state, action, next state, immediate reward, is-s-terminal values in replay buffer; $||.||_2$ is L2 norm. The target of the training process includes: attaining high-quality value estimators, then exploiting these estimators by a policy. Currently, this approach achieves State-Of-The-Art (SOTA) results on a large diverse task set, with algorithms such as DDPG (Lillicrap et al., 2015), TD3 (Fujimoto et al., 2018).

[1]School of Computing and Information Systems, Singapore Management University, 80 Stamford Road, Singapore 178902. Correspondence to: Nguyen Minh Quang <mq.nguyen.2023@phdcs.smu.edu.sg>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024.

In this paper, we identify two weaknesses that contribute to suboptimal policy performance and substantiate our findings with both theoretical and empirical evidences. Firstly, we demonstrate that the value estimator (critic) tends to exploit slowly under Bellman's principle of optimality. We empirically validate this phenomenon in our simulator (**Section 2.2**). Secondly, we highlight sensitivity to critic bias of the policy (actor). To illustrate this, we exert control over non-deterministic factors in our simulator, creating a biased critic (**Section 2.3**). In both scenarios, our analysis of TD3, the current leading algorithm in value-based RL, reveals policies with poor performance.

To address the weaknesses mentioned, we propose enhancing actor and critic (learners) in value-based RL. We incorporate hypothesis representation through two key strategies: (i) implementing weak-model-based learning to improve early exploitation and (ii) employing adaptive rollout to foster adaptability under bias (**Section 3**). The proposed solution functions as an augmenting component integrated into a standard value-based algorithm (**Figure 1**), ensuring the preservation of the inherent advantages of value-based learning. Our approaches effectively tackle the identified challenges, producing noteworthy results across various MuJoCo continuous control tasks (Todorov et al., 2012).

In this work, we make the following contributions:

1. We identify two weaknesses, namely: slow exploitation under Bellman's condition of optimality and bias sensitiveness of value-based learning; then visualize them empirically. We describe the detailed weakness analysis in **Section 2**.

2. We propose Augmenting Learners with Hypothesis (ALH) algorithm which is the combination of: weak-model-learning solution and adaptive rollout solution. Our solutions ensure the exploitation in any estimation, and prevent the bias by a dynamic rollout ability (**Section 3**).

3. We conduct experiments to showcase the improvement in a diverse task set including SOTA benchmarks, and analyze the contribution of each particular component via ablation study on our proposed algorithm. Our algorithm shows a strong improvement in **Section 4**.

2. Problem Analysis

In this section, we present a comprehensive analysis of two weaknesses from both theoretical and empirical viewpoints. We introduce our simulation environment in **Section 2.1**. On our simulation environment, corresponding to each weakness, we explore two distinct schemes:

1. **Non-bias scheme**[1]: No bias control is implemented. We emphasize the explicit manifestation of the **Slow Exploitation** phenomenon, as detailed in **Section 2.2**.

2. **Bias scheme**: We control non-deterministic elements to create bias. We underscore the distinct manifestation of **Bias Sensitiveness**, as expounded in **Section 2.3**.

2.1. Simulated Environment

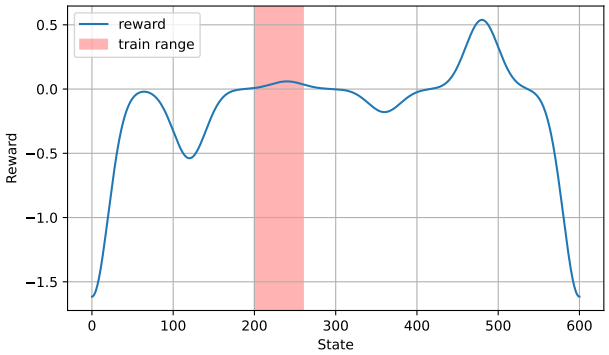

Figure 2. The reward distribution over state space of *MultiNormEnv*. The red span represents initial state sampling in training is limited in a non-representative state range, that can bias the value estimator.

To better monitor the value-based RL agents, we introduce a simple simulation environment named *MultiNormEnv*: $\mathcal{S} = [0, 600]; \mathcal{A} = [-6, 6]; \mathcal{T}(s, a) = s + a$. Given $\Theta = \{(0, -81), (1, -27), (2, 3), (3, -9), (4, 27), (5, -81)\}$, the reward for every tuple of (s, a) relies on the next state (**Figure 2**):

$$n(x, \mu, c) = \frac{c}{20\sqrt{2\pi}} e^{-0.5\left(\frac{x-\mu*120}{15}\right)^2}$$

$$\mathcal{R}(s, a) = \sum_{(\mu,c)\in\Theta} n(\mathcal{T}(s, a), \mu, c)$$

The episode ends if either s' is out of range $[0; 600]$ or the episode length exceeds 100. The optimal policy is expected to frequently travel either of the maximum-reward points, regardless of the initial state.

We train TD3 agents, current SOTA of value-based RL, on two introduced schemes on 10 trials over two million steps. The detailed experiments and benchmarks are described in **Section 4**. To monitor the value of every s' estimated by critics in TD3, we adopt the average Q estimation of two state-action tuples: $(s' - 1, 1), (s' + 1, -1)$. Because the reward only depends on s', the state value curve is expected to have the same shape as the immediate reward.

[1]While the exploration priority, as discussed in the next section, can also be considered a form of bias, we label this scheme as "non-bias" to distinctly delineate between two arguments: weaknesses in the learning scheme and bias induced by the environment.

2.2. Slow Exploitation

Thrun & Schwartz (1993) proved the overestimation on value calculation under Bellman optimization. Following the proof in Thrun & Schwartz (1993), we pay attention to the order between exploitation and exploration in Bellman value-based learning scheme.

Theorem 2.1. *The optimistic value estimator gives higher priority for exploration.*

Proof. We denote the training data (replay buffer) as $\mathcal{D}_{train}$, and $\mathcal{D}_{test}$ is the data on which the agent do rollout. Given action can be ideally sampled, the contribution of action is negligible in uncertainty of estimation. Then, a non-negative $u(s)$ can denote the uncertainty of an estimation on any state s, and $u(s)_{s\sim\mathcal{D}_{train}} \leq u(s)_{s\sim\mathcal{D}_{test}}$. Then, the approximate value made by Q is $Q(s,a) = Q^*(s,a) + \epsilon_u(s)$, where $-u(s) \leq \epsilon_u(s) \leq u(s)$. $\epsilon_u(s)$ can be any noise, Q^* denotes the actual value of Q. Given any $s \in \mathcal{S}$; we define $a_i; s'_i$ are action and next corresponding state that lead to exploration; $a_j; s'_j$ are action and next corresponding seen state. We consider the case that both s'_i and s'_j can lead to same actual value, where $\max_{a\in\mathcal{A}} Q^*(s'_i, a) = \max_{a\in\mathcal{A}} Q^*(s'_j, a)$ and $\mathcal{R}(s, a_i) = \mathcal{R}(s, a_j)$. Denoting $c = \gamma(1-d) \geq 0$, as **Equation 1**, the difference in value estimation between two actions is equal to:

$$\begin{aligned} Q(s,a_i) - Q(s,a_j) &\approx c\left(\max \epsilon_u(s'_i) - \max \epsilon_u(s'_j)\right) \\ &= c\left(u(s'_i) - u(s'_j)\right) \geq 0 \end{aligned} \tag{3}$$

Equation 3[2] shows that optimistic estimator will output equal or higher value for exploration actions than other actions though they share the same actual value. The value estimation is accurate if and only if $u(s'_i) \sim u(s'_j)$, or $\mathcal{D}_{test} \sim \mathcal{D}_{train}$. It indicates that if the number of training steps is not large enough, the output policy is not accurate. □

Empirical Analysis We train TD3 agent with initial states sampled ideally: $s_0 \sim \text{Uniform}(0, 600)$. **Figure 3** expressed approximately the value distribution in the final evaluation. After two million training steps, the non-bias TD3 critics cannot clearly identify all three maximums, therefore cannot accurately reflect the expected values(the green curve). Consequently, TD3 policy is trapped in the first local maximum in more than 1.6 million steps (yellow dots in **Figure 4**).

2.3. Bias Sensitiveness

Remark 2.2. A bias in critic definitely destroys the output policy. As **Equation 2**, the performance of the policy is only decided by the accuracy of the critic.

[2]We provide the detailed proof in **Appendix A.1**.

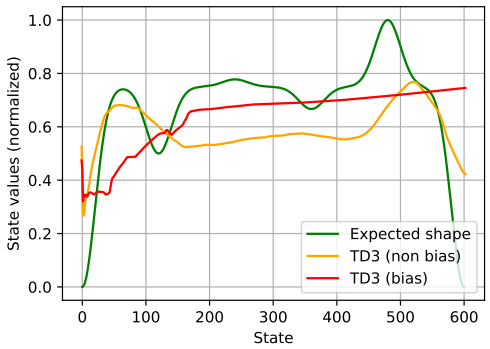

Figure 3. The TD3 state-value estimation after two million training steps in bias and non-bias scheme in *MultiNormEnv*. The reported values are the average of two Q functions, over 10 trials. Values are min-max normalized.

Empirical Analysis To create the bias in the critic, we "pretrained" the critic on biased data and prevent non-value-based discovery. In detail, we apply the following methods on non-deterministic elements. Firstly, as depicted in **Figure 2**, we limit the initial state sampling: $s_0 \sim \text{Uniform}(200, 260)$, and sampled action to zero. Secondly, instead of initializing parameters randomly, we train them on the biased initial replay buffer. Lastly, we remove the exploration noise from policy-based data collecting.

Under this bias, the accuracy of critics is extremely low. As shown in **Figure 3**, the biased TD3 critics show no maximum but the margins of state space. As a result, the policy performance is low. Under the bias, states obtained by TD3 policy distribute uniformly over the whole space (red dots in **Figure 5**), which shows that the policy does not have any target on the state space.

3. Augment Learners with Hypothesis

A good awareness of environment can be the solution for both slow exploitation and possible bias: when the environment model error reflected by the policy decision is decreased, according to Janner et al. (2019), the corresponding performance improvement has higher chance in the true MDP.[3] But defining such a high-quality environment simulator in model-based learning is not always possible. On the other hand, we exploit the concept of hypothesis (**Section 3.1**), which can be incomplete-but-accurate descriptions about environment, then develop related solutions (**Section 3.2** and **Section 3.3**).

3.1. Hypothesis Definition

We define an observation set as a set of tuples of state, action and reward: $o = \{(s_1, a_1, r_1), \dots\} \in \mathcal{O}$; hypothesis space

[3]We provide detailed justification in **Appendix A.2**.

$\mathcal{H}$; metric $\mathrm{D} : \mathcal{O} \times \mathcal{H} \to \mathbb{R}$ measures the difference between a hypothesis and a set of observation o. The hypothesis supported by o is denoted by $\mathrm{M}(o)$ where $\mathrm{M} : \mathcal{O} \to \mathcal{H}$. We proposed to augment the actor/critic with the environment description, $\pi : \mathcal{S} \times \mathcal{H} \to \mathcal{A}$ represents for the actor; $Q : \mathcal{S} \times \mathcal{H} \times \mathcal{A} \to \mathbb{R}$ represents for the critic. We especially pay attention two important characteristics of any hypothesis $h \in \mathcal{H}$, which define how good a hypothesis is:

1. **Validity**: $h \in \mathcal{H}$ is valid if it can explain a set of observation: $\exists o \subset \mathcal{D} | \, \mathrm{D}(o, h) < \mathbb{E}_{o'} \mathrm{D}(\epsilon_o, h)$, where ϵ_o is any non-informative noise. Consequently, a hypothesis can be considered if and only if it is valid.
2. **Robustness**: h_i is more robust than h_j if they are both valid and the supporting set of h_j is a subset of h_i: $o_j \subset o_i$ and $|o_j| < |o_i|$. Equivalently, the more robust a hypothesis is, the larger observation set supports this hypothesis.

The hypothesis validity represents the high exploitation, but does not depend on the value estimation. Concurrently, the dynamicity of inputted hypothesis can result in dynamic policy behavior without changing parameters in model rollout. In the following sections, we introduce certain methods to realize these benefits.

3.2. Weak Model-Based Learning

Achieving the balance condition mentioned in **Equation 3** from early training steps is harmful for exploration. Alternatively, to secure the exploitation, we ensure that there exists non-trivial mapping from value estimation and policy decision to the environment representation. We propose to use an independent model M to capture accurate environment description then feed this information to other learners. Consequently, we can always find the mentioned mappings by reversing the feed-forward processes. Due to M optimization is detached from actor/critic optimization, M does not suffer the priority in exploration.

We name this training process as **weak-model learning**: M can not give strong prediction of the environment interactions, e.g., predicting next state or reward. To simplify notation, M also refers to its learnable parameters. **Figure 4** shows that the policies inputted by the hypothesis (ALH agents) can attain the optimal behavior from very early training steps, in compare with 1.6 million steps of TD3.

Given observation set space $\mathcal{O}$ and $d_{\mathcal{H}} \in \mathbb{N}^*$, $\mathrm{M} : \mathcal{O} \to \mathbb{R}^{d_{\mathcal{H}}}$ is a mapping from observation set to the hypothesis embedding h. In this work, we adopt the Euclidean distance between observation and best recovered output from hypothesis as D:

$$\mathrm{D}(o, h) = \mathbb{E}_{(s,a,r) \in o} \| [s, a, r] - \phi(h, [s, a, r] + \mathcal{N}) \|_2 \quad (4)$$

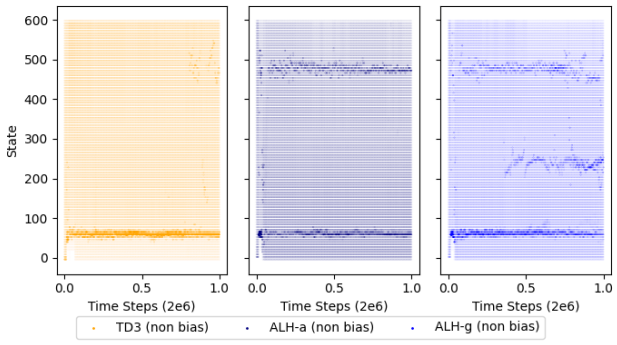

Figure 4. State distributions over time of rollouting TD3, ALH-g and ALH-a policies in evaluation under non-bias scheme. Frequency values are exponentially scaled for visual clarity.

where ϕ is learnable module outputting the denoised observation from hypothesis and noisy observation; $\mathcal{N}$ is Gaussian noise; $[.]$ denotes the concatenate operation. Both M, ϕ are optimized by:

$$(\mathrm{M}^*, \phi^*) = \arg\min_{\mathrm{M}, \phi} \mathbb{E}_{o, h=\mathrm{M}(o)} (\mathrm{D}(o, h) - \sigma * \|h - h_0\|_2) \quad (5)$$

where $o \subset \mathcal{D}_{train}$ is an observation set; σ is a positive hyper-parameter; h_0 is the global hypothesis expected to explain every observation in $\mathcal{D}_{train}$. By adding the diversity coefficient σ, all "local" hypothesis representations are prevented to collapse to a single $h_0 = \mathrm{M}(\mathcal{D}_{train})$.

If only one hypothesis h_0 is used for training actor and critic, overtime, they possibly become overfit. Together with diversity penalty in training M, we propose to construct hypothesis locally for every actor/critic training step: Given o where $|o| = B$ is an observation set for each training step, we sub-sample $\{o_1, \dots o_B\}$ observation sets, each $o_i \subset o$ containing at least one corresponding (s_i, a_i, r_i) observation, and $|o_i| = B_{mini} < B$. Consequently, we can maximize the diversity of the seen hypothesis set based on $\mathcal{D}_{train}$.

3.3. Adaptive Rollout

Our empirical scheme in **Section 2.3** shows that hypothesis is possibly optimized under biased training data. Consequently, the policy will suffer the error from hypothesis representation. As assumption on the robustness in **Section 3.1**, we suggest to align the hypothesis with latest interactions, namely **adaptive rollout** method. Because $\mathcal{D}_{test}$ is unpredictable, the target of this method is to calculate $\mathrm{M}(\mathcal{D}_{train} \cup \mathcal{D}_{test})$ without overhead optimization. **Figure 5** expresses that under bias, the latest-hypothesis-based policies of ALH can attain locally or globally optimal behaviors, even if the critic estimations are not accurate (**Figure 6**-left).

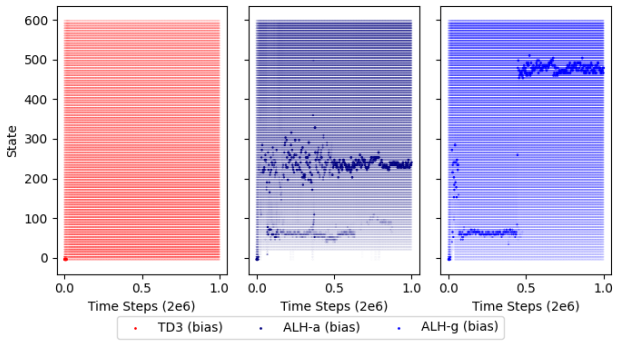

Figure 5. State distributions over time of rollouting TD3, ALH-g and ALH-a policies in evaluation under bias scheme. Frequency values are exponentially scaled for visual clarity.

We employ an incremental learning scheme. For simplification, in this work, we using the add norm function:

$$\mathrm{M}(o_1 \cup o_2) = \mathrm{norm}(\mathrm{M}(o_1) + \mathrm{M}(o_2)) \tag{6}$$

where norm is the Euclidean normalization operation. Mapping to model rollout, o_1 is $\mathcal{D}_{train}$, and o_2 is $\mathcal{D}_{test}$ (or o_{test}). Due to $\mathcal{D}_{train}$ is given, we can optimize h_0 as a parameter directly via **Equation 4**. During model rollout, each decision is made with consideration of the latest hypothesis, as described in **Algorithm 1**.

Algorithm 1 Adaptive rollout

Input: Policy π; Global hypothesis h_0; M is optimized with incremental scheme in 6.
$h'_0 \leftarrow h_0$
for each rollout step **do**
  Run inference with $a \sim \pi(s, h'_0)$
  $o_{test} \leftarrow \{(s_{test}, a_{test}, r_{test})\}$
  $h'_0 \leftarrow \mathrm{norm}(h'_0 + \mathrm{M}(o_{test}))$
end for

We combine two solutions in Augmenting Learners with Hypothesis (ALH) algorithm, presented in **Algorithm 2**. We build ALH on TD3 (Fujimoto et al., 2018)[4]. To simplify the implementation, we attach each ALH training step with a learning agent training step, with a fixed delay of δ_{mem} steps. We consider two discovery methods to collect $\mathcal{D}_{train}$: **ALH-g** in which the policy adopts the most recent **global** h_0 to collect data; **ALH-a** in which the policy adopts the **adaptive** h'_0 hypothesis to collect data, as **Algorithm 1**.

4. Experiments and Results

In the following sections, we present a set of three experiments and corresponding results. Firstly, in our case study

[4]As described in **Appendix B**, ALH can be used as a plug-in with other learning agents.

Algorithm 2 Empirical ALH algorithm

Input: Diversity coefficient σ; Batch size B; Mini batch size $B_{mini} < B$; Replay buffer $\mathcal{D}_{train}$; $\delta_{mem}, \delta_{policy}$.
Initialize critics $Q_{\theta_1}, Q_{\theta_2}$
Initialize actor π_ω
Initialize targets $\theta_{0,1} \leftarrow \theta_1, \theta_{0,2} \leftarrow \theta_2, \omega_0 \leftarrow \omega$
Initialize parameters $\mathrm{M}, \phi, h_0, h'_0 \leftarrow h_0$
for $t = 1$ **to** T **do**
  $\epsilon \sim \mathrm{clip}(\mathcal{N}(0, \tilde{\sigma}), -e, e)$
  Select action $a \sim \pi_\omega(s, h'_0) + \epsilon$, observe reward r and new state s'
  if is ALH-a **then**
    $o_{test} \leftarrow \{(s, a, r)\}$
    $h'_0 \leftarrow \mathrm{M}(h'_0, o_{test})$
    if is done **then**
      $h'_0 \leftarrow h_0$
    end if
  end if
  Add tuple (s, a, r, s') to $\mathcal{D}_{train}$
  Sample a batch of B transitions $o = (s, a, r, s')$ from $\mathcal{D}_{train}$
  if $t \mod \delta_{mem} = 0$ **then**
    Sample $(o_1, o_2) \subset o | o_1 \cap o_2 = \emptyset$
    $h_1 \leftarrow \mathrm{M}(o_1)$
    $h_2 \leftarrow \mathrm{norm}(h_1 + \mathrm{M}(o_2))$
    Update M, ϕ networks using $(o_1, h_1, o_2, h_2, \sigma)$ on **Equation 5**
    Update $h_0 \leftarrow \arg\min_{h_0} \mathrm{D}(o, h_0)$
  end if
  Sample $(o_1, \ldots, o_B)$ from o as **Section 3.2**
  $h' \leftarrow [\mathrm{M}(o_1), \ldots, \mathrm{M}(o_B)]$ (batched)
  $\epsilon \sim \mathrm{clip}(\mathcal{N}(0, \tilde{\sigma}), -c, c)$
  $\tilde{a}_i \leftarrow \pi_{\omega_0}(s'_i, h') + \epsilon$
  $y \leftarrow r + \gamma \min_{i=1,2} Q_{\theta_{0,i}}(s', \tilde{a}, h')$
  Update critics: $\theta_i \leftarrow \arg\min_{\theta_i} \mathbb{E}(y - Q_{\theta_i}(s, a, h'))^2$
  if $t \mod \delta_{policy} = 0$ **then**
    Update actor $\omega \leftarrow \arg\max_\omega \mathbb{E}(Q_{\theta_1}(s, a, h'))$
    Update target networks:
    $\theta_{0,i} \leftarrow \tau\theta_i + (1 - \tau)\theta_{0,i} | i \in \{1, 2\}$
    $\omega_0 \leftarrow \tau\omega + (1 - \tau)\omega_0$
  end if
end for

(**Section 4.2**), we compare our ALH agents against TD3s on *MultiNormEnv* to monitor the critic and actor performance when applying our solutions. To visualize the capabilities of ALH on standard tasks, in the next **Section 4.3**, we compare our ALH against SOTA approaches on a set of eight Mujoco tasks: *Halfcheetah, Hopper, Walker2D, Ant, Humanoid, Reacher, InvertedDoublePendulum*, and *InvertedPendulum*. Lastly, we conduct ablation study on the task *Halfcheetah* in **Section 4.4** in order to monitor the contribution of each component.

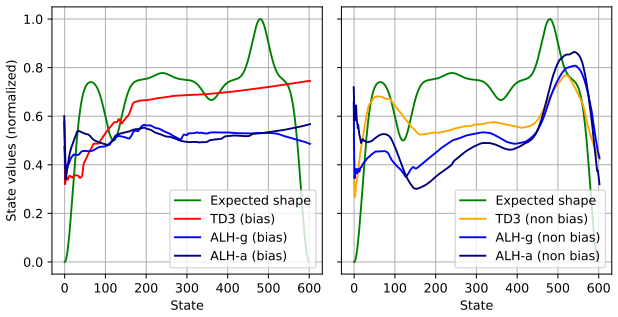

Figure 6. The value estimation of the last training step in bias scheme (left) and non-bias scheme (right). Values are min-max normalized.

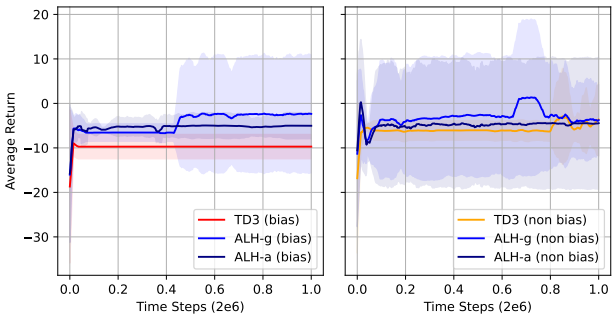

Figure 7. The performance of policies under bias (left) and non-bias (right) in *MultiNormEnv*. The shaded region represents half a standard deviation of the average evaluation over 10 trials. Curves are smoothed uniformly for visual clarity.

4.1. Experiment Setup

Implementation: We utilize the author implementation of TD3[5], MBPO[6] for the performance comparisons. We adopt the DDPG implementation of Fujimoto et al. (2018), and pytorch-based implementation of PPO in Barhate (2021). In all our reported experiments, we use $\delta_{mem} = 10, d_{\mathcal{H}} = 64$, $B_{mini} = \frac{B}{2}, \sigma = 1$. We adopt $B = 256$ in Mujoco tasks[7], and $B = 512$ for a quick coverage in *MultiNormEnv*. For a fair comparison to the learning agent TD3, noise factors $\tilde{\sigma}, e, c$, hyper-parameters δ_{policy}, τ are adopted from TD3 author implementation. With regard to the model size, our implementation utilizes the nearly exact ones in TD3. We run our experiments on Linux environment with 56 CPUs, 8 Nvidia RTX2080Ti.

Time Step Normalization: TD3, DDPG and our work share the same policy update frequency per training step. With regard to **policy update frequency**, the concept of training step between TD3, PPO and MBPO implementation are not computationally equivalent. To make them comparable, our reported step bases on the policy update frequency[8]. In a more detail, our reported one million steps is equivalent to three million steps reported in Schulman et al. (2017), and about first tens thousand steps (depending on particular task) reported in Janner et al. (2019).

Evaluation: There is no data or parameter from evaluation that is reused in training. Each evaluation step contains 10 different episodes. We report the average and the standard deviation across 10 trials, same practice as Fujimoto et al. (2018) and Schulman et al. (2017). We evaluate the policy every 5000 training steps.

[5] https://github.com/sfujim/TD3

[6] https://github.com/jannerm/mbpo

[7] We provide additional results in offline setting in **Appendix C**.

[8] Each training step of MBPO contains significantly high number of policy update steps (up to 40). This number of TD3, DDPG and our work is 0.5. To make the comparison fair, we use the policy update step as training step in MBPO.

4.2. Case Study

We compare our approach, represented by ALH agents, with SOTA value-based approach TD3, under the same conditions mentioned in **Section 2**. The experimental results show the significant improvement over TD3. **Under no bias**: **Figure 6** (right) expresses that ALH agents can highlight minima accurately, and the relative difference between maxima is also closer to the expected than TD3. As a result, ALH's policies can prioritize all maximums as in **Figure 4**. **Under bias**: In **Figure 6** (left), though critics of ALH-a can identify the first two maximums, and critics of ALH-g can uncover the two last, the relative shape differences between them and TD3 critics are unclear. However, **Figure 5** and **Figure 7** express that ALH policies outperform TD3 policies.

In short, our proposed approach can address the weaknesses of slow exploitation and bias sensitiveness of value-based learning scheme. The result shows that ALH can achieve early exploitation, and ALH is less sensitive to the bias than TD3. The improvement is significant in both critic performance and policy performance.

4.3. Mujoco Benchmark

We compare our algorithm against TD3 (Fujimoto et al., 2018); DDPG (Lillicrap et al., 2015) as well as the state of art policy gradient algorithms: PPO (Schulman et al., 2017), and the SOTA of the model-based approach MBPO (Janner et al., 2019) on a set of eight MuJoCo continuous control tasks (Todorov et al., 2012).

In **Figure 8**, our proposed ALH algorithms consistently outperform all baselines across different tasks, showcasing significant improvements over TD3 in terms of overall performance and stability. Value-based learning agents (TD3 and DDPG) exhibit slower discovery speeds in tasks like *Halfcheetah, Hopper, Ant,* and *InvertedDoublePendu-*

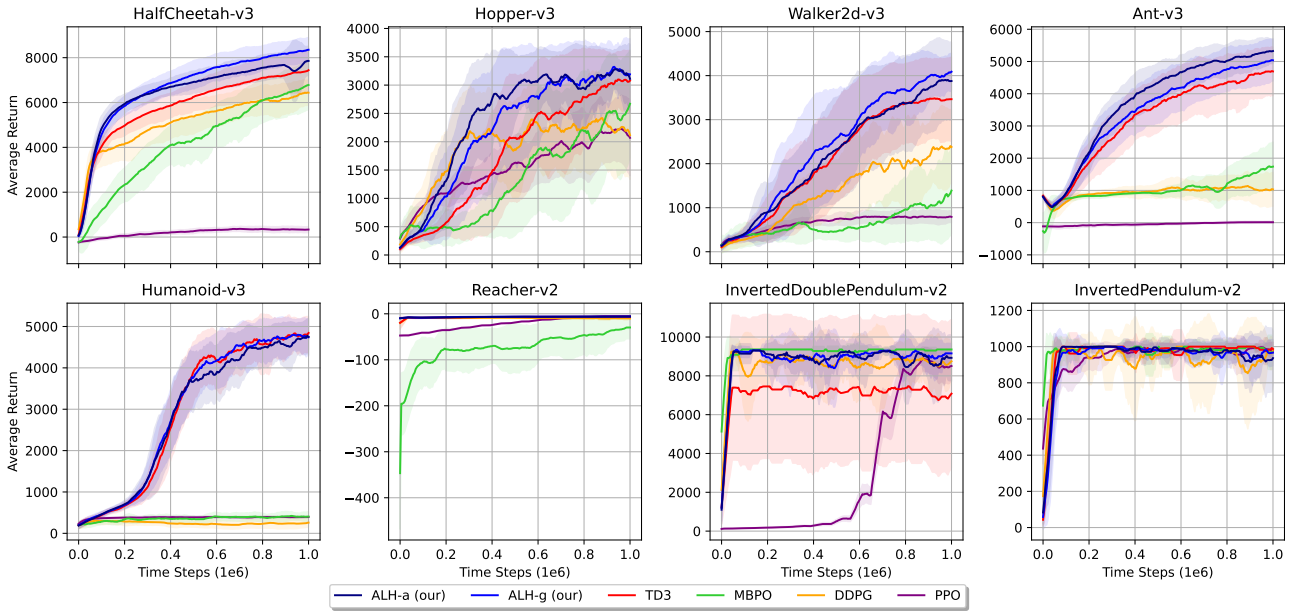

Figure 8. Learning curves for the gym Mujoco continuous control tasks. The shaded region represents half a standard deviation of the average evaluation over 10 trials. Curves are smoothed uniformly for visual clarity.

Environment	ALH-a (our)	ALH-g (our)	TD3	MBPO	DDPG	PPO
HalfCheetah-v3	**7924.09±636.61**	**8508.09±610.20**	7563.45±996.14	6942.36±1108.77	6564.98±517.34	412.87±131.65
Hopper-v3	**3431.29±162.91**	**3439.50±145.63**	3273.77±285.26	2951.81±734.71	2657.49±685.31	2463.91±741.83
Walker2d-v3	**4072.41±827.26**	**4225.65±382.86**	3612.88±860.95	1774.64±1117.82	2698.40±680.68	888.80±237.93
Ant-v3	**5395.55±353.28**	**5179.31±606.92**	4884.83±675.88	1962.42±797.76	1248.39±631.99	22.49±40.47
Humanoid-v3	**5086.73±212.45**	**5036.58±253.53**	5031.64±182.01	475.70±156.08	352.75±95.58	424.07±14.09
Reacher-v2	-5.94±0.53	**-5.79±0.76**	-6.06±0.63	-27.51±24.71	-8.55±0.68	**-5.17±0.37**
InvertedDoublePendulum-v2	9358.61±2.57	**9359.61±0.19**	7492.54±3732.47	**9359.84±0.14**	9358.38±1.87	9094.67±402.77
InvertedPendulum-v2	**1000.00±0.00**	**1000.00±0.00**	**1000.00±0.00**	**1000.00±0.00**	**1000.00±0.00**	**1000.00±0.00**
Avg (normalized)	**85.46**	**86.50**	67.27	36.99	44.70	23.40

Table 1. Average return of the best performed policy. Maximum values of each row are bolded. ± corresponds to a single standard deviation over trials. Normalized score for each policy on a task is 100-scaled min-max normalization across the listed policies.

lum. In other tasks, our method achieves comparable performance, indicating no accuracy trade-off when applying our solutions on value-based learning agent. Moreover, our proposed algorithms consistently achieve the best results, producing the highest average normalized score across all tasks (**Table 1**). In more detail, the two ALH agents achieve the top position in $6/8$ tasks, while taking the runner-ups in the remaining $2/8$ tasks. When compared to TD3, the original learning agent upon which ALH is built, our two algorithms consistently demonstrate remarkable advancement. Additionally, our approach outperforms Model-Based Policy Optimization, while the overhead complexity on the environment model is significantly lower.

To summarize, in all benchmarks, the two ALH agents, consistently outperforms all other baselines. Value-based learning agents (TD3, DDPG) exhibit slower discovery speeds compared to ours. Furthermore, our proposed approach outperforms SOTA of model-based learning (MBPO), and policy-based optimization (PPO).

4.4. Ablation Study

We conduct ablation studies on *HalfCheetah-v3* environment on three variant implementations. In the first variant, namely **no detach**, we update M with the Q and policy objective's gradient. Secondly, in the **no localization** implementation, we use only one hypothesis constructed from current batch[9] for following optimization in a step. Lastly, we disable adaptive rollout by simply not updating the h'_0 during evaluation (**no adaptation**).

Table 2 describes the aggregated returns over one million steps and 10 trials of each mentioned implementations. All proposed methods show significant contribution to the performance improvement, at least four percent and 15 percent

[9]If h_0 is directly utilized, it will encourage learners to ignore never-seen h once h_0 is covered.

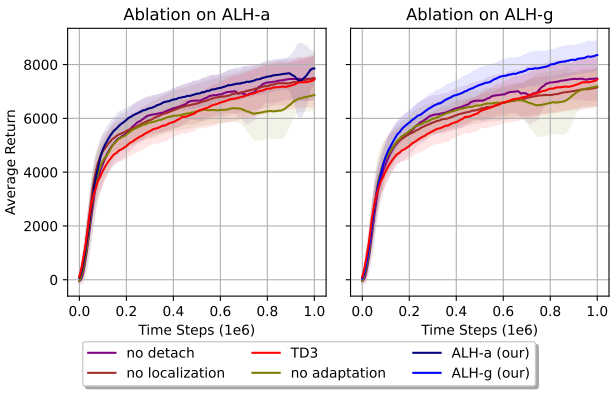

Figure 9. Learning curves for the continuous control *HalfCheetah-v3* task. The shaded region represents half a standard deviation of the average evaluation over 10 trials. Curves are smoothed uniformly for visual clarity.

Aggregate type	Ablation Setting	ALH-g	ALH-a
Max	our	**100%**	**100%**
	no detach	88.94%	95.49%
	no localization	85.00%	96.10%
	no adaptation	85.42%	87.67%
Average	our	**100%**	**100%**
	no detach	91.71%	95.26%
	no localization	88.51%	95.53%
	no adaptation	88.49%	88.43%

Table 2. Ablation study

at most. Adaptive rollout always shows the highest contribution. **Figure 9** expresses that without detaching hypothesis representation, the performance of our proposed ALH-g is approximately equal to TD3, and adaptive rollout shows a significant contribution in the later training steps.

5. Related Work

Generally, there are three main approaches for a RL problem: value-based RL, policy-based RL and model-based RL. In **policy-based RL**, (Mnih et al., 2016) directly optimizes the policy, while Schulman et al. (2015; 2017) indirectly optimizes the target policy with black-box advantage estimation. In contrast, **value-based RL** relies on white-box value estimators, enabling the policy to follow its gradient. This is highlighted by works such as Lillicrap et al. (2015); Fujimoto et al. (2018); Haarnoja et al. (2018). **Model-based RL** generates "imagination experience" to enhance sampling efficiency. Feinberg et al. (2018) partially learns the model by approximating the transition function to improve the value estimator. Meanwhile, Janner et al. (2019) simulates the entire environment, demonstrating that improvements in high-quality imagination experience also translate to the actual environment. To the best of our knowledge, Huang et al. (2022) is the sole work utilizing a "weak-model based" scheme in RL, though the representation of this model lacks clarity without defined criteria.

Adaptation Learning Methods Research in adaptation learning primarily addresses Partial Observable Markov Decision Processes (POMDPs) and offline reinforcement learning tasks, focusing on swift adjustment to changes by leveraging in-context information. Noteworthy works include Chen et al. (2021); Janner et al. (2021) and Ghosh et al. (2022). For POMDPs, Icarte et al. (2020); Demir (2023) equip agents with external memory and control. In a related approach, Jiang et al. (2015) explores "Abstraction Selection" in model-based reinforcement learning. In meta RL, capturing "meta" information (Schmidhuber, 1987), Fakoor et al. (2019); Li et al. (2020) use embedding representations for multiple-task objectives. Additionally, Rakelly et al. (2019) disentangles task inference and control through an additional optimization scheme.

Comparison with Prior Works Weak-model-learning is completely different from model-based reinforcement learning (Janner et al. (2019)), since it does not produce "imagination" experiences. With regard to in-context learning Chen et al. (2021); Janner et al. (2021), the decision in adaptive rollouts relies on a significant wider information. In compare with memory-based methods (Chen et al. (2021); Janner et al. (2021)), hypothesis is independent from policy/critic optimization. In terms of meta-RL, though the design of augmentor M in **Figure 1** is represented similarly as a meta-RL scheme (depicted in Li et al. (2020)), the way we optimize M is completely different. Instead of exploiting the latent useful differences between contexts, we try to incrementally present a large number of observations in training within a task. Consequently, our augmentor is motivated to be a good dynamic observer rather than a crossing-task embedder.

6. Conclusion

In this work, we provide evidence that value-based RL suffers from slow exploitation and bias sensitiveness, from both theoretical and empirical analysis. Based on the idea of model-based RL and adaptation learning, we introduce the Augmenting Learners with Hypothesis solution, to handle the addressed issues. Our approach expresses a outstanding improvement over TD3 algorithm, current SOTA value-based RL algorithm, and also achieves comparative performance with other current SOTAs on a diverse set of benchmarks. As future work, we aim to comprehensively analyze the behavior of ALH on a wider range of different RL algorithms in addition TD3.

Acknowledgements

This research/project is supported by the National Research Foundation, Singapore under its AI Singapore Programme (AISG Award No: AISG2-RP-2021-020).

Impact Statement

This paper presents research aimed at advancing reinforcement learning, specifically addressing low exploitation in early training and bias sensitivity. Although our work has many potential societal implications, we do not feel it is necessary to highlight any specific ones here.

References

Barhate, N. Minimal pytorch implementation of proximal policy optimization. https://github.com/nikhilbarhate99/PPO-PyTorch, 2021.

Chen, L., Lu, K., Rajeswaran, A., Lee, K., Grover, A., Laskin, M., Abbeel, P., Srinivas, A., and Mordatch, I. Decision transformer: Reinforcement learning via sequence modeling. *arXiv preprint arXiv:2106.01345*, 2021.

Demir, A. Learning what to memorize: Using intrinsic motivation to form useful memory in partially observable reinforcement learning. *Applied Intelligence*, pp. 1–19, 2023.

Fakoor, R., Chaudhari, P., Soatto, S., and Smola, A. J. Meta-q-learning. In *International Conference on Learning Representations*, 2019.

Feinberg, V., Wan, A., Stoica, I., Jordan, M. I., Gonzalez, J. E., and Levine, S. Model-based value estimation for efficient model-free reinforcement learning. *arXiv preprint arXiv:1803.00101*, 2018.

Fu, J., Kumar, A., Nachum, O., Tucker, G., and Levine, S. D4rl: Datasets for deep data-driven reinforcement learning, 2020.

Fujimoto, S. and Gu, S. S. A minimalist approach to offline reinforcement learning. *Advances in neural information processing systems*, 34:20132–20145, 2021.

Fujimoto, S., Hoof, H., and Meger, D. Addressing function approximation error in actor-critic methods. In *International conference on machine learning*, pp. 1587–1596. PMLR, 2018.

Ghosh, D., Ajay, A., Agrawal, P., and Levine, S. Offline rl policies should be trained to be adaptive. In *International Conference on Machine Learning*, pp. 7513–7530. PMLR, 2022.

Haarnoja, T., Zhou, A., Abbeel, P., and Levine, S. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International conference on machine learning*, pp. 1861–1870. PMLR, 2018.

Huang, D., Hu, J., Peng, Z., and Hao, M. Weak-model based reinforcement learning control strategy of aircraft attitude system. In *Advances in Guidance, Navigation and Control: Proceedings of 2020 International Conference on Guidance, Navigation and Control, ICGNC 2020, Tianjin, China, October 23–25, 2020*, pp. 5177–5187. Springer, 2022.

Icarte, R. T., Valenzano, R., Klassen, T. Q., Christoffersen, P., Farahmand, A.-m., and McIlraith, S. A. The act of remembering: A study in partially observable reinforcement learning. *arXiv preprint arXiv:2010.01753*, 2020.

Janner, M., Fu, J., Zhang, M., and Levine, S. When to trust your model: Model-based policy optimization. *Advances in neural information processing systems*, 32, 2019.

Janner, M., Li, Q., and Levine, S. Offline reinforcement learning as one big sequence modeling problem. *Advances in neural information processing systems*, 34: 1273–1286, 2021.

Jiang, N., Kulesza, A., and Singh, S. Abstraction selection in model-based reinforcement learning. In *International Conference on Machine Learning*, pp. 179–188. PMLR, 2015.

Kumar, A., Zhou, A., Tucker, G., and Levine, S. Conservative q-learning for offline reinforcement learning. *Advances in Neural Information Processing Systems*, 33: 1179–1191, 2020.

Li, L., Yang, R., and Luo, D. Focal: Efficient fully-offline meta-reinforcement learning via distance metric learning and behavior regularization. In *International Conference on Learning Representations*, 2020.

Lillicrap, T. P., Hunt, J. J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., Silver, D., and Wierstra, D. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*, 2015.

Mnih, V., Badia, A. P., Mirza, M., Graves, A., Lillicrap, T., Harley, T., Silver, D., and Kavukcuoglu, K. Asynchronous methods for deep reinforcement learning. In *International conference on machine learning*, pp. 1928–1937. PMLR, 2016.

Rakelly, K., Zhou, A., Finn, C., Levine, S., and Quillen, D. Efficient off-policy meta-reinforcement learning via probabilistic context variables. In *International conference on machine learning*, pp. 5331–5340. PMLR, 2019.

Schmidhuber, J. *Evolutionary principles in self-referential learning, or on learning how to learn: the meta-meta-... hook*. PhD thesis, Technische Universität München, 1987.

Schulman, J., Levine, S., Abbeel, P., Jordan, M., and Moritz, P. Trust region policy optimization. In *International conference on machine learning*, pp. 1889–1897. PMLR, 2015.

Schulman, J., Wolski, F., Dhariwal, P., Radford, A., and Klimov, O. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Sutton, R. S. and Barto, A. G. *Reinforcement learning: An introduction*. MIT press, 2018.

Thrun, S. and Schwartz, A. Issues in using function approximation for reinforcement learning. *Proceedings of 4th Connectionist Models Summer School, 1993*, 1993.

Todorov, E., Erez, T., and Tassa, Y. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ international conference on intelligent robots and systems*, pp. 5026–5033. IEEE, 2012.

A. Justification

A.1. Exploration Bias

In this section, we provide the detailed justification for our statements in **Section 2.2** about value estimation priority for exploration under Bellman optimality.

Assumption A.1. When action can be easily discovered, e.g. by sampling, once the diversity of action is efficiently high enough to make the action-based estimation confident, we assume the contribution of action is negligible in uncertainty estimation.

Theorem A.2. *The optimistic value estimator gives higher priority for exploration.*

Proof. A typical value-based learning can be defined by the combination of two equations in Q-learning:

$$\theta^* = \arg\min_{\theta} \mathbb{E}_{s,a,s',r,d} ||Q_\theta(s,a) - Q_{target}(s,a)||_2 \tag{7}$$

$$\omega^* = \arg\max_{\omega} \mathbb{E}_s Q(s, \pi_\omega(s)) \tag{8}$$

where $Q_{target}(s,a) = \mathcal{R}(s,a) + \gamma(1-d)\max_{a'} Q(s',a')$, s, a, s', r, d are respectively state, action, next state, immediate reward, is-s-terminal values in replay buffer; $||.||_2$ denotes L2 norm. We denote $V(s) = \max_{a \in \mathcal{A}} Q(s,a)$.

We denote the uncertainty of an estimation on a certain state-action tuple (s,a) by a non-negative $u(s,a)$. As *Assumption A.1*, $u(s,a) \sim u(s) \forall a \in \mathcal{A}$, and $\epsilon_u(s,a) \sim \epsilon_u(s)$. To formally define the difference between known data and new data with regard to the Q function, $u(s)_{s \sim \mathcal{D}_{train}} \leq u(s)_{s \sim \mathcal{D}_{test}}$ ($\sim$ means either belonging or having some "close" features, e.g. negligible L2 distance with a particular record, with the set). The final approximate value made by Q is $Q(s,a) = Q^*(s,a) + \epsilon_u(s)$, where $-u(s) \leq \epsilon_u(s) \leq u(s)$ can be any noise having zero mean, Q^*, V^* denotes the actual value of Q, V respectively.

Given a certain $s \in \mathcal{S}$; we define $a_i; \mathcal{S}'_i = \{s'_i | s'_i = \mathcal{T}(s,a_i)\} | \mathcal{S}'_i \sim \mathcal{D}_{test}$ are any action and corresponding next state set that lead to exploration; $a_j; \mathcal{S}'_j = \{s'_j | s'_j = \mathcal{T}(s,a_j)\} | \mathcal{S}'_j \sim \mathcal{D}_{train}$ are any action and corresponding next state set that lead to examples "similar to" seen data. We consider the case that both $\mathcal{S}'_i$ and $\mathcal{S}'_j$ can lead to same actual value. Given $c = \gamma(1-d) \geq 0$, the priority for exploration can be represented as:

$$\begin{aligned} Q(s,a_i) - Q(s,a_j) &= (\mathcal{R}(s,a_i) + \epsilon_u(s) - \mathcal{R}(s,a_j) - \epsilon_u(s)) + c\left(\max_{a'_i \in \mathcal{A}} Q(s'_i, a'_i) - \max_{a'_j \in \mathcal{A}} Q(s'_j, a'_j)\right) \\ &= (\mathcal{R}(s,a_i) - \mathcal{R}(s,a_j)) + c\left(\max_{a'_i \in \mathcal{A}} (Q^*(s'_i, a'_i) + \epsilon_u(s'_i)) - \max_{a'_j \in \mathcal{A}} \left(Q^*(s'_j, a'_j) + \epsilon_u(s'_j)\right)\right) \end{aligned} \tag{9}$$

Because we are considering the case that both $\mathcal{S}'_i$ and $\mathcal{S}'_j$ can lead to same actual value, where $\mathbb{E}_{a_i} V^*(s'_i) - \mathbb{E}_{a_j} V^*(s'_j) = 0$, and $\mathbb{E}_{a_i} \mathcal{R}(s,a_i) - \mathbb{E}_{a_j} \mathcal{R}(s,a_j) = 0$, then:

$$\begin{aligned} Q(s,a_i) - Q(s,a_j) &= c\left(\max_{a'_i \in \mathcal{A}} (Q^*(s'_i, a'_i) + \epsilon_u(s'_i)) - \max_{a'_j \in \mathcal{A}} \left(Q^*(s'_j, a'_j) + \epsilon_u(s'_j)\right)\right) \\ &\approx c\left(\max_{a'_i \in \mathcal{A}} Q^*(s'_i, a'_i) - \max_{a'_j \in \mathcal{A}} Q^*(s'_j, a'_j)\right) + c\left(\max \epsilon_u(s'_i) - \max \epsilon_u(s'_j)\right) \\ &= c\left(\max_{a'_i \in \mathcal{A}} Q^*(s'_i, a'_i) - \max_{a'_j \in \mathcal{A}} Q^*(s'_j, a'_j)\right) + c\left(u(s'_i) - u(s'_j)\right) \\ &= c\left(V^*(s'_i) - V^*(s'_j)\right) + c\left(\max \epsilon_u(s'_i) - \max \epsilon_u(s'_j)\right) \\ &= c\left(u(s'_i) - u(s'_j)\right) \end{aligned} \tag{10}$$

By the definition of uncertainty u, $c\left(u(s'_i) - u(s'_j)\right) \geq 0$. This inequality shows with the same actual value, optimistic value estimator will give higher estimation for exploration. If and only if $u(s'_i) \sim u(s'_j)$, which means s'_i should be in $\mathcal{D}_{train}$, the estimator outputs accurately. It indicates that **if the number of training steps is not large enough, the output policy is not accurate**. We already expressed the empirical evidence on the non-bias scenario of the prior case study. □

In combination scheme of exploring and exploiting, if we directly limit the value of the unseen data by the value of seen data to secure the exploitation, the learning agent is possibly trapped in a local maximum. This is why, as mentioned, our work does not try to achieve. Then, we do not change the objective functions of actor and critic.

In typical online setting, action sampling enables that in any tuple (s, a), the action a is unlikely to be "unseen". Action sampling can be the result of random initial moves in the replay buffer, or added noise factor in exploration. Building on this insight, we make the assumption that the contribution of the action is negligible in the best uncertainty estimation, leading to $u(s, a) \sim u(s)$ for all $a \in \mathcal{A}$. Then, our proof demonstrates that exploration priority consistently emerges.

Bellman optimality is necessary In combination scheme of exploring and exploiting, if we directly limit the value of the unseen data by the value of seen data to secure the exploitation, the learning agent is possibly always trapped in a local maximum. This is why our work does not try to achieve such balance. Then, we do not change the objective functions of actor and critic.

When *Assumption A.1* is broken? However, this assumption is possibly violated in specific problem classes such as shown our biased scenario and offline settings. In this situation, the theoretical behavior becomes highly unpredictable because both current estimation and future Qs depend on high action-based uncertainty. We consider these situations as bias, but they are not caused by the learning agent's exploration during training. Under a scheme of monotonic dependency on the critic of value-based learning scheme (as **Equation 2**) then static rollout (the decision only bases on learned parameters and input state), the bias in training is intricately reflected. Motivated by minimizing these effects, we propose the second solution, namely adaptive rollout. We empirically demonstrate this improvement in both our case study in **Section 2.3** and in offline RL of **Appendix C**.

A.2. Justification for ALH

In this section, we give detailed justification for our method. We begin with relaxing the concept of environment model error from the context of model-based learning by the introduction of the "average reconstruct method" ρ^{-1}. Then, we utilize ρ^{-1} for analyzing the behaviour of plain Bellman learning agents. Lastly, we explain the two ideas behind weak-model learning and adaptive rollout respectively. During our explanation, we also explain terms mentioned in the main pages.

Assumption A.3. In continuous domain, the probability of a prediction is proportional to the inverse of L2 distance between the prediction and the true observation.

Assumption A.4. A non-trivial reconstruct method is defined as a generator outputting the data/pattern used to train a certain approximator; and the probability of this data/pattern must be non-zero. **The goodness of an approximator can be measured by the probability of the average reconstructed pattern from this approximator.** To avoid the distribution definition[10], we adopt the average method, whose goodness is the expectation of all non-trivial reconstruct methods.

Derive policy improvement in simulation setting As presented in *Theorem 4.1.* of MBPO paper (Janner et al., 2019), when the returns under the model: $\hat{\eta}[\pi]$ ($\equiv J(\pi)$ on artificial environment) by **more than** $\mathbf{C}(\epsilon_\mathbf{m}, \epsilon_\pi)$, the improvement is guaranteed in the actual environment. Here, ϵ_m is the environment simulation error, ϵ_π is the distribution shift between behavior policy and our evaluated policy. Since $C \propto \epsilon_m$, on the same other conditions, π having lower value ϵ_m is expected to perform better on actual environment, or $\epsilon_m[\pi_1] > \epsilon_m[\pi_2] \iff J(\pi_1) > J(\pi_2)$.

Since ϵ_m in MBPO paper is limited by the context of simulation setting, we derive it to the general context by analysing the output of optimized learners. Denoting ρ is the optimizer, a denotes an optimized learner: $a = \rho(f)$, where f is a dynamic[11]. On the reverted side, $\exists\ \rho^{-1} : \{a\} \mapsto \{\hat{f}\}$ where $\hat{f}$ is the **reconstructed dynamic** from a ($\mathrm{dom}\hat{f} = \mathrm{dom}f$)[12]. As *Assumption A.4*, simulation accuracy can be presented by the probability $P(\rho^{-1}(a) \equiv f) \leq P(f \equiv f) = 1$. As *Assumption A.3*, $P(\rho^{-1}(\pi) \equiv \mathcal{M}) \propto \frac{1}{\epsilon_m[\pi]}$. In plain Bellman optimization, Q is the only instruction of π: $\exists \rho, \rho_1, \rho_2$ such that $\pi = \rho_1(Q)$ where $Q = \rho_2(\mathcal{M})$, or $\pi = \rho(\mathcal{M}) = \rho_1(\rho_2(\mathcal{M}))$. Equivalently, the model error of π is cumulative error

[10] We can adopt the definition of a distribution of all reconstruct methods (stochastically), and the goodness based on KL divergence instead of the deterministic prediction probability; which also brings the same result, but a bit more complicated. In future work, we will develop a comprehensive theory.

[11] Equivalently the complete training data distribution.

[12] In MBPO (Janner et al., 2019), ρ^{-1} is the learnable simulation dynamic.

made by simulating model $\mathcal{M}$ by Q and/then simulating Q by π. Assume that ρ_1, ρ_2 are independent:

$$\begin{aligned} P(\rho^{-1}(\pi) \equiv \mathcal{M}) &= P(\rho_2^{-1}(\rho_1^{-1}(\pi)) \equiv \mathcal{M}) = P(\rho_1^{-1}(\pi) \equiv Q) * P(\rho_2^{-1}(Q) \equiv \mathcal{M}) \\ &\Rightarrow P((\rho^{-1}(\pi)) \equiv \mathcal{M}) \leq P(\rho_2^{-1}(Q) \equiv \mathcal{M}) \\ &\Rightarrow \epsilon_m[\pi] \geq \epsilon_m[Q] \end{aligned}$$

However, as we proved, early Bellman Qs are likely overestimated. Q optimization is also empirically proved to heavily suffer bias effect. Both situations end up with high value of $\epsilon_m[Q]$. **To sum up, policy is early poor performed due to the misalignment with the environment model, as a result of plain Bellman optimization and single dependency on Q instructions.**

Alternative instruction As discussed, we can expect the high performance of π when its $\epsilon_m[\pi]$ is the lowest, or:

$$\begin{aligned} &\epsilon_m[\pi] = \min_a \epsilon_m[a] \\ &\Rightarrow \exists \rho_3^{-1}, \rho_4^{-1}, h = \arg\min_a \epsilon_m[a] \mid \rho_3^{-1}(\pi) \equiv \rho_4^{-1}(h) \\ &\Rightarrow \exists f_2 \mid \pi = f_2(h) \end{aligned}$$

where f_2 is any non-trivial mapping. In another word, when π is constrained by the term h ($\pi = f_2(h)$); its performance improvement is guaranteed regardless of Q accuracy. To compare with plain Bellman optimization, the improvement is visible when $\epsilon_m[h] < \epsilon_m[Q]$. Such f_2 is the mentioned awareness of environment in **Section 3**. We find that f_2 will be automatically guaranteed to be non-trivial in h-augmented setting ($\pi \equiv \pi(.|h)$): since any optimizer maximizing $J(\pi)$[13] needs to keep $\epsilon_m[\pi(.|h)] \approx \epsilon_m[h]$; the contribution of h in $\pi(.|h)$ must not be trivial. Consequently, this non-trivial contribution is namely f_2.

Adaptability is necessary However, such mapping also have a problem: optimal h is available if and only if we have complete knowledge of dynamic $\mathcal{M}$. Consequently, our h is always an approximation, thus possibly trapped in a local optimal due to the training distribution. 1) In terms of optimization problem, the policy performance is possibly trapped when only one "global" h is adopted. 2) With regard to inference improvement direction, denoting rollouting data by $\mathcal{D}_{test}$; the corresponding $\epsilon_m[h_{\text{roll}}]$ of $\mathcal{D}_{train} \cup \mathcal{D}_{test} \subset \rho^{-1}(h_{\text{roll}})$ must be at least equal or smaller than $\epsilon_m[h_{\text{train}}]$ of $\mathcal{D}_{train} \subset \rho^{-1}(h_{\text{train}})$, since $\mathcal{D}_{test} \sim \mathcal{M}$ and $|\mathcal{D}_{train} \cup \mathcal{D}_{test}| \geq |\mathcal{D}_{train}|$. Equivalently, policy of hypothesis h_{roll} has higher probability to perform good then this of h_{train}, which is optimized on available $\mathcal{D}_{train}$. To sum up, **a diverse set of hypothesis is necessary for policy optimization, and adaptive inference using $\mathcal{D}_{test}$ helps to increase the probability of good policy performance.**

Hypothesis definition To describe h conceptually, we introduce the definition of hypothesis, where a single hypothesis represent for a certain environment "possibility". Particularly, we define an observation set as a set of tuples of state, action and reward: $o = \{(s_1, a_1, r_1), \dots\} \in \mathcal{O}$ (We discuss the reason for this incomplete definition in **Appendix A.3**); hypothesis space $\mathcal{H}$; metric $\mathrm{D} : \mathcal{O} \times \mathcal{H} \to \mathbb{R}$ measures the difference between a hypothesis and a set of observation o. The hypothesis supported by o is denoted by $\mathrm{M}(o)$ where $\mathrm{M} : \mathcal{O} \to \mathcal{H}$. We proposed to augment the actor/critic with the environment description, $\pi : \mathcal{S} \times \mathcal{H} \to \mathcal{A}$ represents for the actor; $Q : \mathcal{S} \times \mathcal{H} \times \mathcal{A} \to \mathbb{R}$ represents for the critic. **Recall the mentioned criteria for goodness in Section 3, the higher validity is equivalent to lower visible part of $\epsilon_m[h]$ [14]; while the higher robustness implies the lower invisible part of $\epsilon_m[h]$.**

A.3. Incomplete Observation Definition

We define an incomplete observation of (s, a, r) in **Section A.2**, instead of the complete tuple of (s, a, r, s', d). Additionally, we also adopt a very high noise scale in denosing task (the noise scale is 20 percent in each input dimension); use very sparse update frequency of weak-model objective (**Algorithm 2**) and adopt two-layer M, D networks. The motivation for the definition and configurations is **to prevent the weak-model from becoming too strong**. As we argued in **Section A.2**, h only need to be $\epsilon_m[h] < \epsilon_m[Q]$ then the improvement on the policy is visible. By analysing the Bellman error in early

[13] Recall our argument why Bellman optimization is important in previous section, beside the automatic uncertainty-based exploration, Bellman optimization also aims to maximizing $J(\pi)$.

[14] Being visible here means we can only validate $\epsilon_m[h]$ partly on known observation record.

training state and also the empirical results, our argument is proved. In the future work, we aim to comprehend the analysis of these factors.

A.4. Toy Environment Design

We introduced environment *MultiNormEnv* for the purpose of visualizing learning agent in continuous state/action setting. There are some motivations for such design:

1. **Domain** As we limit the scope of our work is continuous space; state and action spaces are continuous range.
2. **Visualization** We opt for single-dimension state/action for visualization clarity purpose. To avoid the complex correct $V^*(s) = \max_a Q^*(s, a)$, we design the reward only rely on the value of next state. As a result, value function must be proportional to the shape of reward function.
3. **Non-trivial** We utilize a particular value set of Θ to create different local optimums. As our design, a biased observation set can not generalize the whole environment distribution.

B. ALH as a Plug-in

The introduced ALH in **Algorithm 2** is equivalent to an implementation on top of TD3 algorithm with two discovery strategies. TD3 is currently the SOTA algorithm for a large continuous environment benchmark set, we reported the ALH built on TD3 for close comparisons and clean presentation. Our ALH algorithm can be separated from the learning agent as an independent module. As our definition, ALH solutions have no dependency on the learning agent. To show this argument, we present the theoretical version of ALH by **Algorithm 3**, in which our solutions are free from the detail implementation of learning agent $\mathcal{L}$.

Algorithm 3 Theoretic ALH

Input: Learning Agent $\mathcal{L}$; Diversity coefficient σ; Batch size B; Mini batch size $B_{mini} < B$; Replay buffer $\mathcal{D}_{train}$
{Optimize weak model}
Initialize parameters M, ϕ, h_0
for each ALH train step **do**
  Sample $(o_1, o_2) \subset \mathcal{D}_{train}$
  $h_1 \leftarrow \mathrm{M}(o_1)$; $h_2 \leftarrow \mathrm{norm}(h_1 + \mathrm{M}(o_2))$
  Update M, ϕ networks using $(o_1, h_1, o_2, h_2, \sigma)$ on **Equation 5**
  Update $h_0 \leftarrow \arg\min_{h_0} \mathrm{D}(o_1 \cup o_2, h_0)$
end for
{Localize hypothesis construction for training $\mathcal{L}$}
for each $\mathcal{L}$ train step **do**
  Sample $(o_1, \ldots, o_B)$ as **Section 3.2**
  $(h_1, \ldots, h_B) \leftarrow (\mathrm{M}(o_1), \ldots, \mathrm{M}(o_B))$
  Optimize $\mathcal{L}$ w.r.t $(h_1, \ldots, h_B)$
end for
{Adaptive rollout}
$h'_0 \leftarrow h_0$
for each rollout step **do**
  Run inference $\mathcal{L}$ with h'_0
  $o_{test} \leftarrow \{(s_{test}, a_{test}, r_{test})\}$
  $h'_0 \leftarrow \mathrm{norm}(h'_0 + \mathrm{M}(o_{test}))$
end for

To show our argument about a plug-in implementation and also as a stronger evidence that the two addressed problems are not specific for TD3, we conduct experiment on DDPG algorithm. Since the Q feedback is the only instruction for the policy, DDPG also relies on plain Bellman optimization to update the policy. We conduct our experiment on 8 introduced Mujoco environments on 10 trials, under the same hyper-parameters setting and running conditions. **Figure 10** compares learning curves of our ALHs with DDPG. ALH also shows significant improvements over DDPG: the improvement is witnessed in

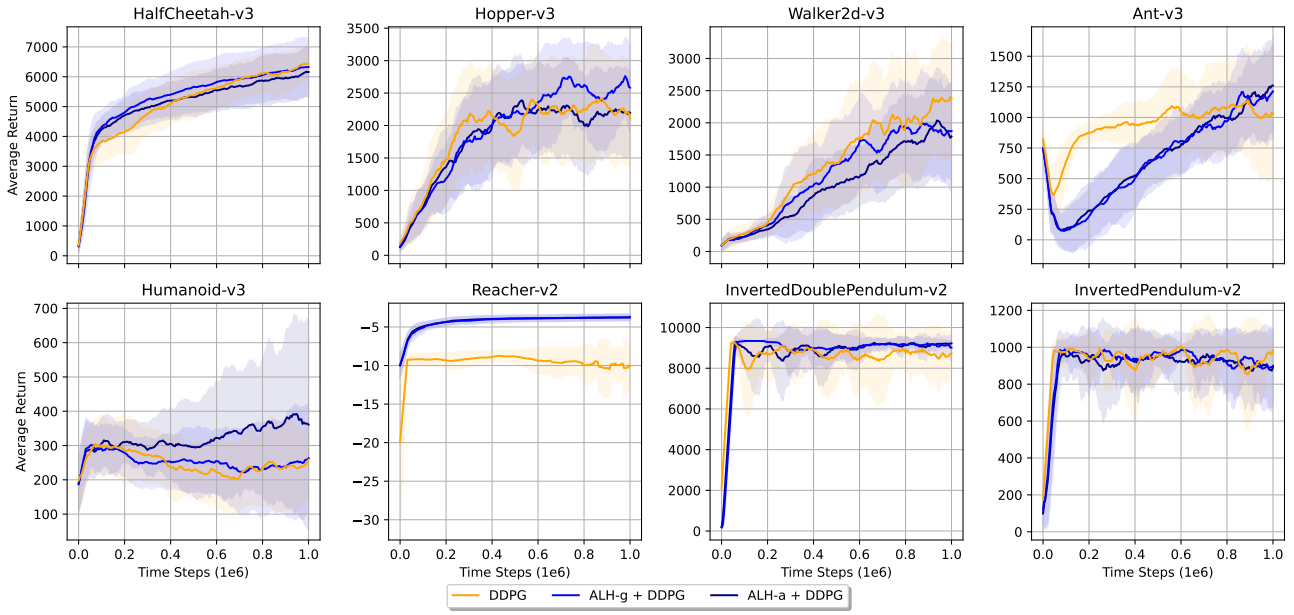

Figure 10. Learning curves for the gym Mujoco continuous control tasks. Our ALHs are implemented on the top of DDPG. The shaded region represents half a standard deviation of the average evaluation over 10 trials. Curves are smoothed uniformly for visual clarity.

4/8 environments, while performance is closed to DDPG in the remaining. Especially, in *Humanoid* and *Reacher*, ALH agents show very different performance pattern with DDPG: it is not trapped or decreasing.

C. Promising in Offline Setting

Offline setting is a typical case of bias, due to no exploration is available. In this setting, we compare our proposed implementations against TD3+BC[15] and Behavior Cloning[16] (BC), two strong baseline in offline RL, together with reported results of CQL (Kumar et al., 2020), DT (Chen et al., 2021) and APV-E (Ghosh et al., 2022) for their original papers on continuous control datasets from the D4RL benchmark (Fu et al., 2020). We observe that no much different between trials, so we report our proposed, TD3+BC and BC on only one trial, but use a total of 50 episodes on 5 different seeds for each evaluation step, over one million training steps. Specifically, we conducted experiments on two ALH variants:

- **ALH**: We normalize the observations while only removing the discovery part of the online setting from **Algorithm 2**.
- **ALH+BC**: Following Fujimoto & Gu (2021), we add a behavior cloning penalty to the policy objective function with the same coefficient to the former implementation.

Table 3 reports the best performed policies on *halfcheetah*, *hopper*, *walker2d* D4RL environment with five increasing quality data versions: *random, medium, medium-replay, medium-expert, expert*. Reported score in offline setting is normalized as same as Fu et al. (2020). Our original ALH algorithm outperforms in random-collected datasets but witness a significant drop in higher-quality datasets. On the other hand, our proposed ALH+BC always defeats the performance of behavior cloning baseline, and can improve or match the performance of the SOTA TD3+BC in a high probability: $9/15$ datasets witness the improvement; and two datasets witness the approximate performance (total 11 over 15). In compare with other baselines, our proposal often outperforms CQL and APE-V; and can approximately meet the performance of reported results of DT.

Table 4 describes the aggregated average returns on mentioned runs of each our proposed algorithms in offline setting. The reported results of online setting is on one million train steps, on five D4RL *"halfcheetah"* datasets. Generally, all solutions

[15] https://github.com/sfujim/TD3_BC

[16] We simplify remove the critic-based optimization from the implementation of TD3+BC.

Environment	ALH+BC (our)	ALH (our)	TD3+BC	BC	DT	APE-V
halfcheetah-random-v2	17.30±0.07	**35.85±0.14**	14.88±0.02	2.26±0.00	-	29.9
hopper-random-v2	9.88±0.02	**34.51±0.14**	32.30±0.00	8.05±0.45	-	31.3
walker2d-random-v2	7.40±0.16	**20.27±0.11**	**21.95±0.00**	5.75±0.39	-	15.5
halfcheetah-medium-v2	49.96±0.47	47.91±2.30	48.81±0.30	43.09±0.26	42.6	**69.1**
hopper-medium-v2	66.88±1.41	1.78±0.00	**74.77±4.09**	63.58±2.25	67.6	-
walker2d-medium-v2	86.84±0.34	1.55±0.24	85.68±0.95	77.80±1.27	74.0	**90.3**
halfcheetah-expert-v2	**98.90±0.34**	3.75±0.15	**97.81±0.36**	93.81±0.24	-	-
hopper-expert-v2	**112.95±0.37**	1.59±0.00	**112.46±0.04**	**112.61±0.18**	-	-
walker2d-expert-v2	**111.21±0.09**	1.87±0.28	**110.99±0.05**	**109.36±0.10**	-	-
halfcheetah-medium-expert-v2	96.12±0.27	10.53±1.19	96.55±0.40	74.53±4.47	86.8	**101.4**
hopper-medium-expert-v2	**112.74±0.18**	1.68±0.00	**112.26±0.23**	89.40±8.10	107.6	105.72
walker2d-medium-expert-v2	**111.29±0.07**	0.21±0.00	**111.32±0.13**	**109.45±0.16**	108.1	**110.0**
halfcheetah-medium-replay-v2	46.55±0.23	49.55±3.56	45.46±0.42	38.51±0.81	36.6	**64.6**
hopper-medium-replay-v2	**100.78±2.08**	**99.90±0.15**	**99.12±1.00**	78.88±6.56	82.7	98.5
walker2d-medium-replay-v2	**90.64±1.05**	76.26±3.89	87.60±0.50	54.77±3.14	66.6	82.9
Avg All	**74.63**	25.81	**76.80**	64.12	-	-
Avg Reported	74.63	25.81	**76.80**	64.12	74.73	72.66

Table 3. Average return of the best performed policy. Maximal values of each row are bolded. ± corresponds to a single standard deviation over evaluation seeds.

show contribution in enhancement, and detaching hypothesis optimization hold the most important role. Meanwhile, localization and adaptation possibly have negative contribution in performance.

Aggregate type	Ablation Setting	ALH+BC	ALH
Max	our	100%	100%
	no detach	36.92%	2.56%
	no localization	99.43%	**114.78%**
	no adaptation	**101.43%**	100.63%
Average	our	**100%**	**100%**
	no detach	30.98%	-5.92%
	no localization	99.70%	88.31%
	no adaptation	97.98%	98.62%

Table 4. Ablation study on offline setting

To explain the observation on original ALH: having promising performance on random data but poor performance on higher-quality data, using a policy to collect data should be categorized as a strong bias. This scenario limits the hypothesis diversity, therefore M is overfitted. It worth noting that we still use Bellman optimization, in combination with inability to describe the environment, policy is encouraged to do out-of-history exploration. Though adding BC has promising performance on high-quality data, it depends on altering the policy (and indirectly the critic), which also relies on the behavior policy quality. In our future work, we aim to comprehensively address this problem.

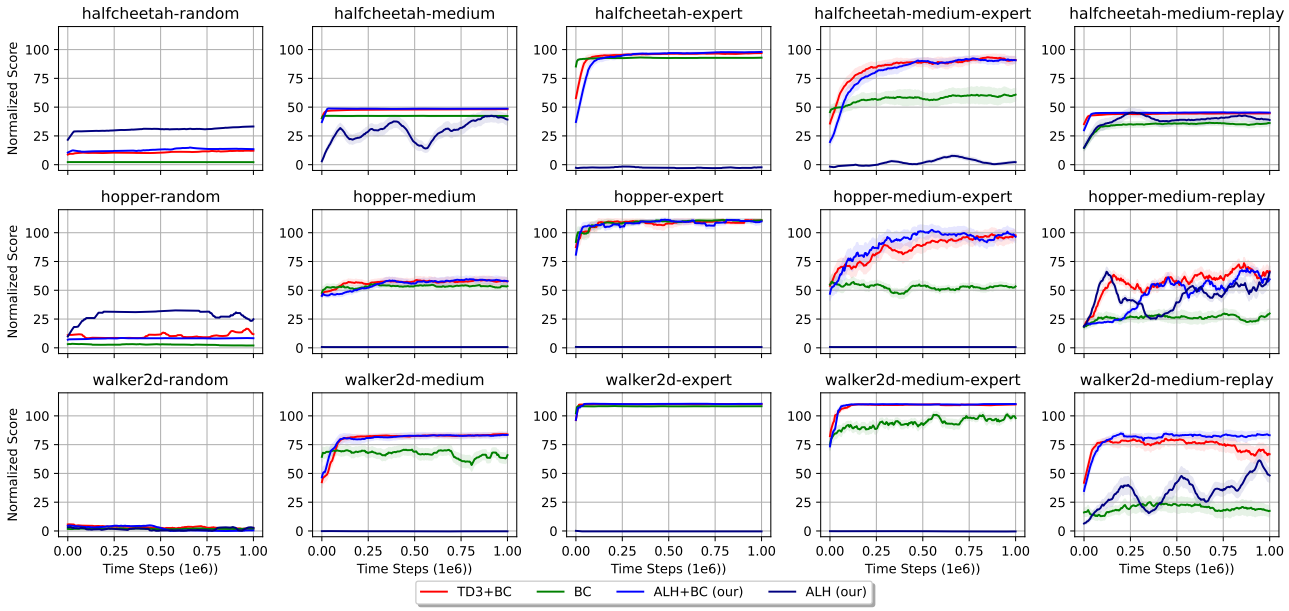

Figure 11. Offline learning curves. The shaded region represents half a standard deviation of the average evaluation over 5 evaluation seeds. Curves are uniformly smoothed for visual clarity.

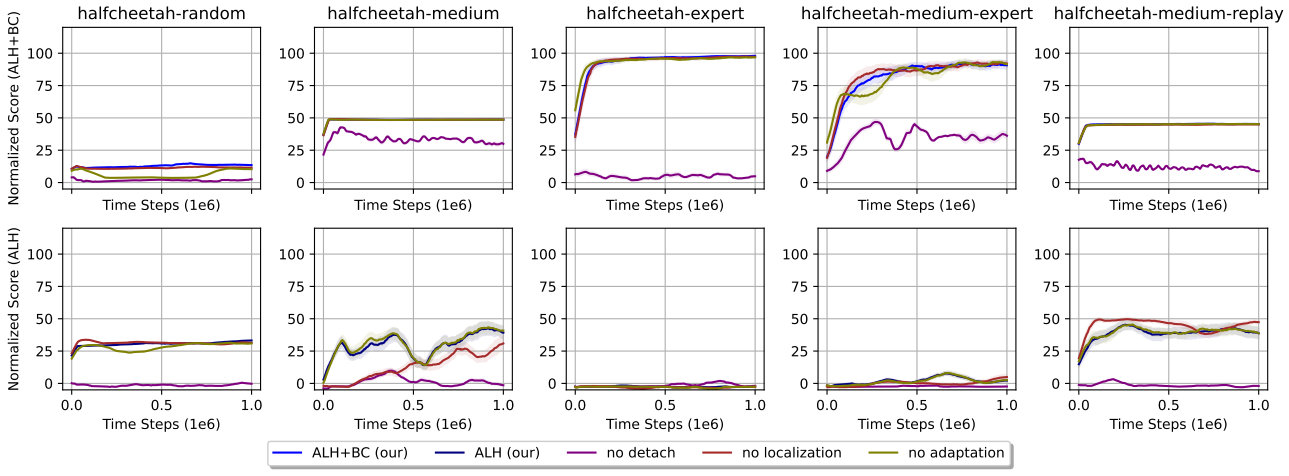

Figure 12. Offline ablation learning curves. The shaded region represents half a standard deviation of the average evaluation over 5 evaluation seeds. Curves are uniformly smoothed for visual clarity.