A RELATED WORK

In this work, we consider the exploration strategy under the principle of *Optimism in the Face of Uncertainty* (OFU) (Auer et al., 2002), especially in the heteroscedastic stochastic environment. We aim to improve exploration efficiency, and alleviate the over exploration issue caused by aleatoric uncertainty.

Basic exploration strategies, like ϵ-greedy (Sutton & Barto, 2018), noise perturbation (Lillicrap et al., 2016), entropy regularization (Mnih et al., 2016) and stochastic policy (Haarnoja et al., 2018), lead to undirected exploration through random perturbations. With the increasing emphasis on exploration efficiency in RL, various exploration methods have been developed. One kind of methods uses intrinsic motivation to stimulate agent to explore, such as count-based novelty (Martin et al., 2017; Ostrovski et al., 2017; Bellemare et al., 2016; Tang et al., 2017; Fox et al., 2018), prediction error (Pathak et al., 2017), reachability (Savinov et al., 2019) and information gain on environment dynamics (Houthooft et al., 2016). Some recent methods, originating from tracking uncertainty, guide efficient exploration under the principle of OFU, such as Thompson Sampling (Thompson, 1933; Osband et al., 2016), IDS (Nikolov et al., 2019; Clements et al., 2019) and other customized methods (Moerland et al., 2017; Pathak et al., 2019).

The base of OFU methods is to model epistemic and aleatoric uncertainties in RL. Bootstrapped DQN (Osband et al., 2016) has become the well-used approach for capturing epistemic uncertainty (Kirschner & Krause, 2018; Ciosek et al., 2019), and distributional RL methods (Bellemare et al., 2017; Zhou et al., 2020; Dabney et al., 2018a;b) are used for capturing aleatoric uncertainty. However, most traditional OFU methods do not distinguish the two types of uncertainty, which can easily lead the naive solution to favor actions with higher variances in stochastic tasks, i.e., over-exploration issue.

To address that, Mavrin et al. (2019) study how to take advantage of value distribution for efficient exploration under both types of uncertainty, proposing Decaying Left Truncated Variance (DLTV) based on QR-DQN. Besides, Nikolov et al. (2019) and Clements et al. (2019) propose to use Information Direct Sampling (Kirschner & Krause, 2018) for efficient exploration in RL, which formulates epistemic and heteroscedastic aleatoric uncertainty and maximizes information gain on globally optimal action to explore informative state-action pairs. However, such methods are complicated when deriving a behavior policy and is limited to discrete control.

Meanwhile, there is not any strategy that can help the well-performed continuous RL algorithms (Haarnoja et al., 2018; Ciosek et al., 2019; Ma et al., 2020) to address aleatoric uncertainty when exploration. OAC (Ciosek et al., 2019) proposes exploration bonus guided by the upper bound of Q estimation to facilitate exploration based on Soft Actor-Critic (SAC) (Haarnoja et al., 2018). Nevertheless, OAC ignores the potential impact of the aleatoric uncertainty, which may cause misleading exploration. Our proposed OVD-Explorer is a novel exploration strategy, which can guide agent to explore towards higher epistemic uncertainty, and also avoid the areas with high aleatoric uncertainty, improving the robustness of exploration especially facing heteroscedastic aleatoric uncertainty.

To capture aleatoric uncertainty, OVD-Explorer models the value distribution and uses mutual information to guide exploration following the principle of OFU, measuring the correlations between the policy distribution and upper bounds distribution of return. There are some other information-theoretic exploration strategies using mutual information, such as VIME (Houthooft et al., 2016), which measures the information gain on environment dynamics, and EMI (Kim et al., 2019), which generates intrinsic reward using prediction error of representation learned by mutual information. Those methods can solve sparse reward problem very well by using intrinsic reward. Nevertheless, those exploration methods use mutual information neither on the value distribution, nor for OFU-based exploration. Besides, unlike the mechanisms used in measuring mutual information, such as variational inference (Hinton & van Camp, 1993; Houthooft et al., 2016) and f-divergence (Nowozin et al., 2016; Kim et al., 2019), we find the correlation between policy and upper bounds of return through uncertainty as shown in Theorem 1, thus we can directly derive the close form exploration policy.

B PROOFS

B.1 PROOF OF THEOREM 1

In order to prove the Theorem 1, we first propose the following lemma about $\mathbf{F}^\pi(s)$.

Lemma 1. *The mutual information of $\bar{Z}^\pi(s,a_0),\cdots,\bar{Z}^\pi(s,a_k)$ and $\pi(s)$ at state s is:*

$$\mathbf{F}^\pi(s)=\sum_{a\sim\pi(a|s)}\mathop{\mathbb{E}}_{\bar{z}(s,a)\sim\bar{Z}^\pi(s,a)}\left[p(a|\bar{z}(s,a),s)\log\frac{p(a|\bar{z}(s,a),s)}{\pi(a|s)}\right], \tag{18}$$

where $p(a|\bar{z}(s,a),s)$ represents the posterior probability distribution of policy given current state s and the sampled upper bound of return $\bar{z}(s,a)$.

Proof. For simplicity, we assume that $k=2$, and derive the mutual information among three random variables $\mathbf{MI}(\bar{Z}^\pi(s,a),\bar{Z}^\pi(s,a'),\pi(s))$, where $a\neq a'$.

Considering the formula for the mutual information, $\mathbf{F}^\pi(s)$ is derived as follows:

$$\begin{aligned}
\mathbf{F}^\pi(s)&=\mathbf{MI}(\bar{Z}^\pi(s,a),\bar{Z}^\pi(s,a'),\pi(s)|s)\\
&=\sum_{\substack{a\sim\pi(s)\\ \bar{z}(s,a)\sim\bar{Z}^\pi(s,a)\\ \bar{z}(s,a')\sim\bar{Z}^\pi(s,a')}}\left[p(a,\bar{z}(s,a),\bar{z}(s,a'))\log\frac{p(a,\bar{z}(s,a),\bar{z}(s,a'))}{\pi(a|s)p(\bar{z}(s,a),\bar{z}(s,a'))}\right]\\
&=\sum_{\substack{a\sim\pi(s)\\ \bar{z}(s,a)\sim\bar{Z}^\pi(s,a)\\ \bar{z}(s,a')\sim\bar{Z}^\pi(s,a')}}\left[p(a|\bar{z}(s,a),\bar{z}(s,a'))p(\bar{z}(s,a),\bar{z}(s,a'))\log\frac{p(a|\bar{z}(s,a),\bar{z}(s,a'))}{\pi(a|s)}\right]\\
&=\sum_{a\sim\pi(s)}\mathop{\mathbb{E}}_{\substack{\bar{z}(s,a)\sim\bar{Z}^\pi(s,a)\\ \bar{z}(s,a')\sim\bar{Z}^\pi(s,a')}}\left[p(a|\bar{z}(s,a),\bar{z}(s,a'))\log\frac{p(a|\bar{z}(s,a),\bar{z}(s,a'))}{\pi(a|s)}\right],
\end{aligned}$$

where the posterior distribution $p(a|\bar{z}(s,a),\bar{z}(s,a'))$ is the probability of choosing action a on the condition of the samples from upper bounds of action a and a'.

Considering that in the decision-making process, the probability of action a is independent to the upper bound of other actions, which means that $p(a|\bar{z}(s,a),\bar{z}(s,a'))=p(a|\bar{z}(s,a))$. Therefore, $\mathbf{F}^\pi(s)$ can be further reduced as follows.

$$\mathbf{F}^\pi(s)=\sum_{a\sim\pi(s)}\mathop{\mathbb{E}}_{\bar{z}(s,a)\sim\bar{Z}^\pi(s,a)}\left[p(a|\bar{z}(s,a),s)\log\frac{p(a|\bar{z}(s,a),s)}{\pi(a|s)}\right]. \tag{19}$$

□

Lemma 1 tells that the mutual information $\mathbf{F}^\pi(s_t)$ is in direct proportion to $p(a|\bar{z}(s,a),s)$, which measures how much it is worth acting under the current policy $\pi(a|s)$ when the upper bound is known.

Next, to measure the posterior probability $p(a|\bar{z}(s,a),s)$, we use a general and practically effective approach (Wang & Jegelka, 2017; Belakaria et al., 2020; Perrone et al., 2019; Li et al., 2020) of approximating the posterior probability given upper bound value.

Specifically, we approximate $p(a|\bar{z}(s,a),s)$ using the prior that $z^\pi(s,a)\leq\bar{z}(s,a)$ with given policy $\pi(s,a)$, since $\bar{z}(s,a)$ is the upper bound of $z^\pi(s,a)$. Hence, we use the indicator function $1_{z^\pi(s,a)\leq\bar{z}(s,a)}$ to truncate the policy $\pi(s,a)$, and utilize the constant C to normalize the probability, as is shown in the following equation.

$$p(a|\bar{z}(s,a),s)=\frac{1}{C}\pi(a|s)\mathbb{E}_{z^\pi(s,a)\sim Z^\pi(s,a)}\left[\mathbb{1}_{z^\pi(s,a)\leq\bar{z}(s,a)}\right].$$

Here, $\mathbb{E}_{z^\pi(s,a)\sim Z^\pi(s,a)}\left[\mathbb{1}_{z^\pi(s,a)\leq\bar{z}(s,a)}\right]=\Phi_{Z^\pi(s,a)}(\bar{z}(s,a))$, where Φ_x is the cumulative distribution function (CDF) of x, $\bar{Z}^\pi$ and Z^π are the random variables, whose distributions describe the randomness of the returns, and $\bar{z}(s,a)$ is the value of random variable $\bar{Z}^\pi$.

Therefore, the posterior probability can be measured as follows,

$$p(a|\bar{z}(s,a),s) = \frac{1}{C}\pi(a|s)\Phi_{Z^\pi}(\bar{z}(s,a)). \tag{20}$$

In our method, we do not use the commonly used mechanisms about mutual information such as neural network estimation (Belghazi et al., 2018) and upper bound estimation (Cheng et al., 2020). Instead, we can find the correlation between random variables as shown in Equation 20, which helps to derive mutual information directly.

According to Lemma 1 and Equation 20, we can give the proof of Theorem 1 in the following.

Proof. By Combining Lemma 1 and Equation 20, $\mathbf{F}^\pi(s)$ can be further derived as follows.

$$\begin{aligned}
\mathbf{F}^\pi(s) &= \sum_{a\sim\pi(s)} \mathop{\mathbb{E}}_{\bar{z}(s,a)\sim\bar{Z}^\pi(s,a_0)} \left[p(a|\bar{z}(s,a),s) \log \frac{p(a|\bar{z}(s,a),s)}{\pi(a|s)} \right] \\
&= \sum_{a\sim\pi(s)} \mathop{\mathbb{E}}_{\bar{z}(s,a)\sim\bar{Z}^\pi(s,a)} \left[\frac{1}{C}\pi(a|s)\Phi_{Z^\pi}(\bar{z}(s,a)) \log \frac{\pi(a|s)\Phi_{Z^\pi}(\bar{z}(s,a))}{C\pi(a|s)} \right] \\
&= \sum_{a\sim\pi(s)} \mathop{\mathbb{E}}_{\bar{z}(s,a)\sim\bar{Z}^\pi(s,a)} \left[\frac{1}{C}\pi(a|s)\Phi_{Z^\pi}(\bar{z}(s,a)) \log \frac{\Phi_{Z^\pi}(\bar{z}(s,a))}{C} \right] \\
&= \frac{1}{C} \mathop{\mathbb{E}}_{\substack{a\sim\pi(s) \\ \bar{z}(s,a)\sim\bar{Z}^\pi(s,a)}} \left[\Phi_{Z^\pi}(\bar{z}(s,a)) \log \frac{\Phi_{Z^\pi}(\bar{z}(s,a))}{C} \right]
\end{aligned}$$

Here, the last equality follows from Theorem 1. □

B.2 Proof of proposition 1

Proof. Similar to (Ciosek et al., 2019), we set the covariance matrix of behavior policy π_E is that of target policy π_T, i.e., $\Sigma_E = \Sigma_T$. Hence, the OVD-Explorer problem is simplified as:

$$\begin{aligned}
\mu_E &= \arg\max_{\mu} \hat{\mathbf{F}}(s) \\
&= \arg\max_{\mu,} \mathop{\mathbb{E}}_{\bar{Z}^\pi} \left[\Phi_{Z^\pi}(\bar{z}(s,\mu)) \log \frac{\Phi_{Z^\pi}(\bar{z}(s,\mu))}{C} \right]
\end{aligned} \tag{21}$$

To ensure that the behavior policy samples actions around the target policy, we derive the π_E upon mean μ_T of target policy π_T. In specific, we firstly obtain the gradient of $\hat{\mathbf{F}}(s,\mu)$ at π_T, which is given as follows:

$$\nabla_a \hat{\mathbf{F}}^\pi(s,\mu)|_{\mu=\mu_T} = \mathbb{E}_{\bar{Z}^\pi} \left[\hat{m} \times \frac{\partial \bar{z}(s,a)}{\partial a}|_{a=\mu_T} \right] \tag{22}$$

where $\hat{m}$ is given as:

$$\hat{m} = \phi_{Z^\pi(s,\mu_T)}(\bar{z}(s,\mu_T))(\log \frac{\Phi_{Z^\pi(s,\mu_T)}(\bar{z}(s,\mu_T))}{C} + 1), \tag{23}$$

and $\phi(x)$ is the probability distribution function (pdf). Hence, the μ_E is given as follows:

$$\mu_E = \mu_T + \alpha \mathbb{E}_{\bar{Z}^\pi} \left[m \times \frac{\partial \bar{z}(s,a)}{\partial a}|_{a=\mu_T} \right], \tag{24}$$

where α is the step size controlling exploration level and $m = \log \frac{\Phi_{Z^\pi(s,\mu_T)}(\bar{z}(s,\mu_T))}{C} + 1$. □

C Algorithm 2: OVD-Explorer for DSAC

In this section, we show the whole algorithm of our implementation of OVD-Explorer based on DSAC in Algorithm 2. All the code can be found in the supplementary material.

Algorithm 2 OVD-Explorer for DSAC

Initialise: Value networks θ_1, θ_2, policy network ϕ and their target networks $\bar{\theta}_1$, $\bar{\theta}_2$, $\bar{\phi}$, quantiles number N, target smoothing coefficient (τ), discount (γ), an empty replay pool $\mathcal{D}$
for each iteration **do**
  for each environmental step **do**
    $a_t \sim \pi_E(a_t, s_t)$ according to Algorithm 1
    $\mathcal{D} \leftarrow \mathcal{D} \cup \{(s_t, a_t, r(s_t, a_t), s_{t+1})\}$
  end for
  for each training step **do**
    for i = 1 to N **do**
      for j = 1 to N **do**
        calculate $\delta^k_{i,j}, k = 1, 2$, following Eq. 4
      end for
    end for
    Calculate $\mathcal{L}_{QR}(\theta_k), k = 1, 2$ using $\delta^k_{i,j}$ following Eq. 2
    Update θ_k with $\nabla \mathcal{L}_{QR}(\theta_k)$
    Calculate $\mathcal{J}_\pi(\phi)$, following Eq. 5
    Update ϕ with $\nabla \mathcal{J}_\pi(\phi)$
  end for
  Update target value network with $\bar{\theta}_k \leftarrow \tau\theta_k + (1-\tau)\bar{\theta}_k, k = 1, 2$
  Update target policy network with $\bar{\phi} \leftarrow \tau\phi + (1-\tau)\bar{\phi}$
end for

Table 4: OVD-Explorer parameters

	Parameter		Value
Training	Discount		0.99
	Target smoothing coefficient	τ	5e-3
	Learning rate		3e-4
	Optimizer		Adam (Kingma & Ba, 2015)
	Batch size		256
	Quantiles amount		20
	Replay buffer size		1.0×10^6 for Mujoco tasks
			1.0×10^5 for other tasks
	Environment steps per epoch		1.0×10^3 for Mujoco tasks
			1.0×10^2 for other tasks
Exploration	Exploration ratio	α	0.05
	Uncertainty ratio	β	3.2
	Normalization factor	C	0.5

D MORE DETAILS ABOUT THE EXPERIMENTS

D.1 GRIDCHAOS

GridChaos is an environment built on OpenAI's Gym toolkit, whose map is shown as in Figure 3(a) and Figure 5. In this section we illustrate more details in addition to Section 5.2.

The movable cyan triangle and the fixed symmetric dark blue goal are two parts split from square, and the goal is to make the triangle embedded in the goal to recover the original square, which is to say that it is an isosceles triangle whose base side is equal to the height. The triangle is always initialised randomly in the cyan rectangle, and the black line in the map represents the wall, where the triangle will be adsorbed once hits the wall. The state transition is stochastic, and we add Gaussian noise to the action resulting in Gaussian transition probability.

To represent the location of the triangle, we establish a Cartesian coordinate system using the centroid of the map as the origin, as shown in Figure 5. Then the coordinates of the triangle are represented by the midpoint of the altitude of the triangle which is shown as the red point in the triangle. In the case shown in Figure 5, the initial coordinate of the triangle (agent) is in the negative half of the x-axis and the task target is in the first quadrant.

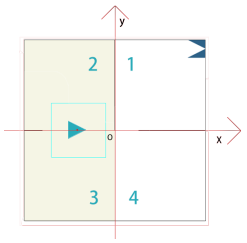


Figure 5: Map of GridChaos.

D.2 Baselines

Ma et al. (2020) shows the performance of DSAC and TD4, which is the distributional extension of TD3 (Fujimoto et al., 2018), and can also be used to capture epistemic and aleatoric uncertainty. Moreover, DSAC outperforms TD4 on Mujoco tasks as shown in Ma et al. (2020), so we evaluate only on SAC and DSAC, and further implement OVD-Explorer based on DSAC to develop the exploration ability.

SAC. The SAC (Haarnoja et al., 2018) implementation is mainly based on OAC repository, and the results in Ant-v2 and Hopper-v2 are similar to reported results by OAC. Our SAC report a better result than OAC's implementation for SAC on HalfCheetah-v2, which is because the high variance of this environment as explained as in OAC.

DSAC. The DSAC (Ma et al., 2020) implementation is based on our implementation of SAC, except that the distributional Q function in the DSAC repository is used instead of the traditional Q function in SAC. As it is based on SAC, we set the hyper-parameters of DSAC to be consistent with SAC to ensure the fair comparison, which also results in the different reported results from original paper of DSAC. In our results, DSAC can guarantee an absolute advantage over SAC in most cases, which is consistent with the previous conclusion.

DOAC. The DOAC implementation is mainly based on our implementation of DSAC as well as the open source code of OAC. As DSAC shows great advantage due to the distributional value estimation, to ensure a fair comparison, we extend OAC (Ciosek et al., 2019) to its distributional version, i.e., DOAC, by replacing the exploration process of DSAC by the behavior policy derived by OAC. We set the hyper-parameters the same as used by OAC in Mujoco[3], and our results of DOAC on Ant-v2 and HalfCheetah-v2 are significantly better than that OAC reported.

D.3 Implementation

Our implementation of OVD-Explorer is based on the open source code of OAC [4], also refer to the code of DSAC [5] as well as softlearning [6]. All experiments are performed on NVIDIA GeForce RTX 2080 Ti 11GB graphics card.

The training process of OVD-Explorer and DOAC are the same as in DSAC, except for the different behavior policy used, while OVD-Explorer and DOAC enrich the experience replay with the data using the the derived exploration policies, respectively. To ensure the fair comparison, the hyper-parameters for training process of baselines and OVD-Explorer are the same. Besides, we have three hyper-parameters associated with OVD-Explorer as mentioned before, including α that controls the exploration level, β that determines the magnitude of uncertainty we use, as well as C that is the normalization factor. The hyper-parameters in our experiments are shown in Table 4.

E More Experiment Study

E.1 Runtime analysis

[3] That is given by the open source code, where β_{UB} is 4.66 and δ is 23.53

[4] https://github.com/microsoft/oac-explore

[5] https://github.com/xtma/dsac

[6] https://github.com/rail-berkeley/softlearning

Figure 6 shows the time consumption of algorithms relative to SAC. As can be seen, the distributional value estimation used in DSAC, DOAC and our methods introduces extra time consumption distinctly. Nevertheless, the relative time consumption of OVDE_G and OVDE_Q to SAC is 1.21 and 1.17, respectively, which means that OVD-Explorer spends about 20% more time than SAC to achieve up to nearly 100% performance gain as shown in Figure 6(b). This demonstrates the extra time consumption is well worth it. Besides, the time consumption of OVDE_Q is close to that of DSAC, only with a larger variance, which indicates that the additional time consumption of OVDE_Q is minimal while performing better exploration.

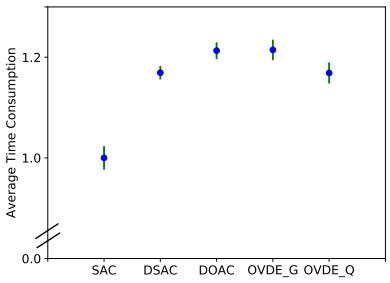
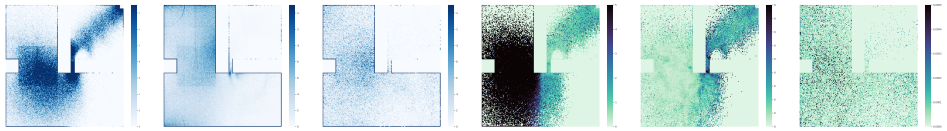


Figure 6: Runtime analysis. The data is from 1 trial of each algorithm on Noisy Ant-v2 task, and the errorbar represents half of the standard deviation.

E.2 Analysis about OVD-Explorer's advantage in the case of GridChaos

With the heatmap of the visiting frequency of agent during exploration, and the heatmap about the uncertainty estimation, we can visually analyze the patterns and advantages of OVD-Explorer.



(a) (b) (c) (d) (e) (f)

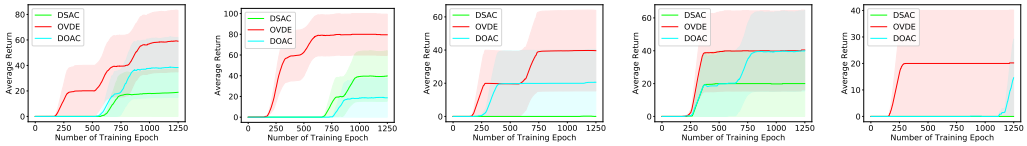
Figure 7: State visiting frequency heatmap from 1.0×10^5 to 2.5×10^5 steps of one trial for (a) OVD-Explorer, (b) DSAC and (c) DOAC. (d) Estimated aleatoric uncertainty of OVD-Explorer; (e) Epistemic-aleatoric ratio of OVD-Explorer; (f) Estimated uncertainty for exploration in DOAC.

The distinctly different exploration patterns can be easily found. Figure 7(a), 7(b) and 7(c) present the state visiting frequency of OVD-Explorer, DSAC and DOAC, respectively. We can see that OVD-Explorer explores directly to the right half, where the environmental randomness is lower, whereas DSAC and DOAC are both stuck in the left half with higher environmental randomness.

Furthermore, we show how OVD-Explorer could explore directly without being trapped by the randomness through the estimated uncertainty. In specific, Figure 7(d) shows that the aleatoric uncertainty estimated by OVD-Explorer is consistent with environment settings, where the environment noise is higher on the left half. Figure 7(e) shows the ratio of estimated epistemic uncertainty and aleatoric uncertainty (i.e., epistemic-aleatoric ratio) of OVD-Explore, and higher ratio means higher epistemic uncertainty or lower aleatoric uncertainty, which is exactly the direction OVD-Explorer explores . The ratio is larger on the right half, which means that OVD-Explorer can avoid being stuck in the left half. Meanwhile, Figure 7(f) presents the estimated uncertainty in DOAC, which is larger on the left half. As DOAC encourages exploring area with relatively large estimated uncertainty, it explains why DOAC is stuck in the left half.

E.3 Evaluation on several other noise scale in GridChaos

OVD-Explorer can explore efficiently in heteroscedastic stochastic environment by considering differently about epistemic and aleatoric uncertainty for exploration as shown in Section 5.2. To further empirically prove its strength, we test OVD-Explorer in GridChaos with different noise scales in four quadrants, and the result is shown as Figure 8 and Table 5. We can find that OVD-Explorer can perform well in all those different noise injection of environment.



(a) (b) (c) (d) (e)

Figure 8: Training curves on GridChaos with noise of different scale. The x-axis indicates number of training epoch (100 environment steps for each training epoch), while the y-axis is the evaluation result represented by average episode return. The shaded region denotes half standard deviation of average evaluation over 5 seeds. Curves are smoothed uniformly for visual clarity. These results are corresponding to row *A* to row *E* in Table 5.

Table 5: The results for GridChaos. Noise setup shows the different setup for environmental heterogeneous Gaussian noise scale, and the corresponding four columns represent the noise settings in the four quadrants of the Cartesian coordinate system, as shown in Figure 5, in which the goal locates in the first quadrant, and the triangle is initialised in the second or third quadrants. Average return shows the average episodic undiscounted return with half standard derivation in the last 100 epoch before totally 1250 epochs. FRG epoch means the minimum training epoches in the trials used to *Firstly Reach the Goal* before totally 1250 epochs. The row *S* is the standard GridChaos as shown in Figure 3(b), the others are shown in Figure 8

	Noise setup				Average return			FRG epoch		
	1	2	3	4	DSAC	OVDE	DOAC	DSAC	OVDE	DOAC
S	0.1	0.5	0.5	0.1	0.00±0.00	**59.30**±48.42	3.02±6.03	1250+	**229**	1222
A	0.0	0.5	0.1	0.1	18.94±37.87	**58.99**±48.18	38.42±47.08	1161	**180**	662
B	0.0	0.05	0.01	0.01	39.78±48.72	**79.52**±39.77	18.71±37.41	694	**144**	846
C	0.05	0.1	0.1	0.05	0.05±0.10	**39.64**±48.55	20.59±39.66	1250+	**180**	309
D	0.001	0.005	0.005	0.001	20.00±40.00	**40.46**±48.61	39.99±48.97	284	**276**	321
E	0.0	0.0	0.0	0.0	0.00±0.00	**20.20**±39.90	14.60±29.20	1250+	**185**	1118

Concretely, from the results, the following observations deserve to be noticed. First, OVD-Explorer can significantly achieve better average return in all those settings, especially when the noise is set high, and can learn to reach the goal faster (see column about FRG). It shows the ability of OVD-Explorer to guide agent explore against higher aleatoric uncertainty on the left side (the second and third quadrants). Second, for the task without noise as shown in row *E*, which means the state transition is deterministic, OVD-Explorer still learns quickly. The results in row *E* show the inherently high difficulty of this task, not only because of the very sparse reward, but also the gate leading to the goal is set very small (the width of the gate is only 30% of the length of agent, i.e., the base of the isosceles triangle, which means that at the doorway the agent can only move a very small distance horizontally, otherwise it would be adsorbed to the wall and immobile).

E.4 Evaluation in GridChaos when the aleatoric uncertainty is high around goal

In the previous experiments in GridChaos, the noise (i.e., aleatoric uncertainty) near the goal is set lower. In such situation, OVD-Explorer, which follows the principle of OFU and further avoids exploring areas with higher aleatoric uncertainty, could bring significant advantage. Such setup of heterogeneous noise is reasonable, because in real life, the goal or optimal policy is always not expected to be highly stochastic.

Nevertheless, the evaluation about tasks with the existence of high randomness in the target region is valuable, so we conducted the following experiment in GridChaos, where the environment randomness in the right half (first and fourth quadrants), where the target is located, was set larger. The results are shown in Table 6. Note that we use OVDE(P) to denote the usual implementation that pessimistically estimates the value distribution (i.e., using Equation 13). Besides, OVDE(M) denotes the implementation that does not pessimistically estimate the value distribution (i.e., we modify the

mean of Gaussian distribution Z^π in Equation 13 from the lower bound to expected value of the Q estimation $\mu(s, a)$ as in Equation 11.).

Table 6: The results for GridChaos (additional). This shows row F to J, which are the cases where the optimal policy would face higher aleatoric uncertainty.

	NOISE SETUP		AVERAGE RETURN				FRG EPOCH			
	1&4	2&3	DSAC	OVDE(P)	OVDE(M)	DOAC	DSAC	(P)	(M)	DOAC
F	0.5	0.1	**50.10**±40.96	16.61±33.22	17.01±33.86	**50.52**±41.43	**479**	1071	583	**321**
G	0.1	0.05	0.00±0.00	19.84±39.68	**39.96**±48.95	20.14±39.93	-1	**188**	226	233
H	0.05	0.005	0.00±0.00	**20.69**±39.61	**20.07**±39.94	0.00±0.00	- 1	**247**	308	-1
I	0.01	0.005	0.0±0.0	40.00±48.99	**60.00**±48.99	39.99±48.98	-1	**200**	236	301
J	0.005	0.001	20.00±40.00	**39.98**±40.97	20.00±40.00	20.00±40.00	236	312	**200**	296

Our experimental findings are mainly the following three aspects.

Firstly, when facing extremely high aleatoric uncertainty around the goal (see row F), which causes the interaction around goal to be very unstable, chaotic and disorder, OVD-Explorer would strongly discourage exploring such a area, and thus performance would be damaged. In contrast, DSAC and DOAC have no restriction on aleatoric uncertainty, and high randomness may instead increase the probability of achieving the goal.

Second, in most cases (see G, H, I, J), OVD-Explorer always can guide better exploration and achieve better performance than DSAC and DOAC, especially when the noise is negligible (see row J). This reflects the fact that our exploration objective (the mutual information shown in Equation 8) makes great sense, achieving an appropriate trade-off between avoiding high aleatoric uncertainty and being optimistic about high epistemic uncertainty.

Third, an interesting finding is that OVD-Explorer may perform better by turning off the pessimistic estimation facing higher aleatoric uncertainty around the goal (see column OVDE(M)). This suggests that excessive pessimism is unnecessary if there is a need to explore areas with high aleatoric uncertainty.

Overall, from the results in Table 5 and Table 6, OVD-Explorer is able to tackle most of the cases quite well. When there is high randomness around the goal, OVD-Explorer has a shortcoming that it will inevitably slow down the efficiency of reaching the goal, because it limit the exploration towards such area. Fortunately, this shortcoming can be mitigated by turning off the pessimistic estimation.

E.5 EVALUATION OF STATISTICAL SENSE

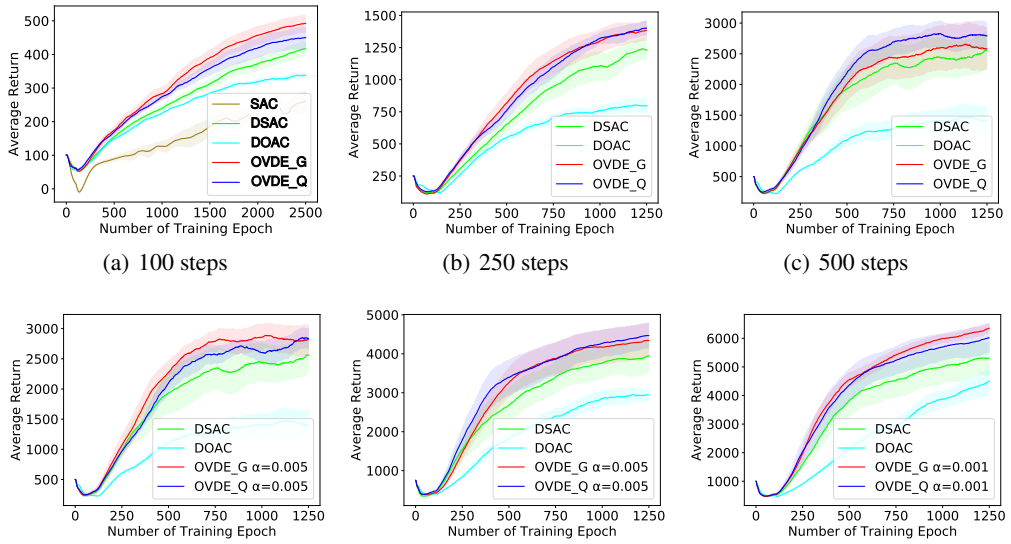
Table 7: Comparisons of related algorithms on Ant-v2. We report the averaged performance and standard deviation.

TASK	EPOCH	SEED	DSAC	DOAC	OVD-EXPLORER_G	OVD-EXPLORER_Q
ANT-V2	2500	0, 1, 2, 3, 4	6206.9±1202.5	6586.7±1023.3	7160.6±763.2	**7590.3**±154.9
ANT-V2	2500	5, 6, 7, 8, 9	6565.0±1343.0	6664.2±255.5	**7190.1**±813.8	7174.3±570.0
ANT-V2	2500	0 - 9	6385.9±1287.2	6625.4±7446.8	7175.3±789.0	**7382.3**±466.6

To counteract the randomness from a statistical perspective, we conduct all experiments for 5 trails with different seeds (typically 0-5), and report the average results with standard deviation. Next, to verify that the 5 trails are sufficient to mitigate the statistical randomness, we run other 5 runs (seeds are set as 5, 6, 7, 8, 9, respectively) for those algorithms on Ant-v2, and show the results in the following. Note that the first row of results is from our previously reported results, which is the same as Table 3, and the second row show the results new. The experimental results in Table 7 show that the results of 5 trials are sufficiently representative of the overall level, while the performance of OVD-Explorer undoubtedly stays ahead.

E.6 STUDY ON EPISODIC HORIZON

Section 5.3 has shown great advantage of OVD-Explorer over DSAC and DOAC in stochastic Mujoco tasks, which limits the length for an episode to 100 steps. To further empirically verify the efficiency of OVD-Explorer, we test on Noisy Ant-v2 task with different maximum episodic length setup. Our results in Figure 9 show that OVD-Explorer can significantly perform better than baselines in different maximum episodic length (i.e., 250, 500, 750 and 1000). Noting that longer maximum episodic length renders higher difficulty of solving tasks, especially for the high-dimensional tasks demanding exploration. In specific, we have the following two conclusions.



(a) 100 steps (b) 250 steps (c) 500 steps

(d) 500 steps (e) 750 steps (f) 1000 steps

Figure 9: Training curves on Noisy Ant-v2 tasks with different maximum episodic length setup. The x-axis indicates number of training epoch (the number of environment steps for each training epoch is the same as the episodic horizon), while the y-axis is the evaluation result represented by average episode return. The shaded region denotes half standard deviation of average evaluation over 5 seeds. Curves are smoothed uniformly for visual clarity. The sub-title of each figure represents the episodic horizon, also known as the maximum episode length.

Firstly, the exploration should be more conservative in the harder tasks, where we should set smaller α in OVD-Explorer. In Figure 9(a), (b) and (c), α is set to 0.05 by default, while we can find that the advantage of OVD-Explorer_G decreases gradually with the increasing of the difficulty of tasks. Further, if α is set to 0.005, then there is a substantial performance improvement as shown in Figure 9(d). Also, as shown in Figure 9(e), both OVD-Explorer methods perform well when the task episodic horizon is 750 with α set to 0.005. On the hardest task we tested, i.e., the Noisy Ant-v2 with horizon 1000 as shown in Figure 9(f), OVD-Explorer gain remarkable performance, while α is set smaller as 0.001.

Secondly, OVD-Explorer_Q is more stable than OVD-Explorer_G, which is consistent with the conclusion in Section 5.3. We can find from Figure 9(a), (b) and (c) that OVD-Explorer_Q performs stably better while OVD-Explorer_G degrades. OVD-Explorer_G is better in easier task with horizon 100, which is due to the Gaussian prior of noise. But when the task becomes harder, the prior helps less, and OVD-Explorer_Q shows the advantage of more flexibly modeling aleatoric uncertainty and thus the performance is more stable.

E.7 STUDY ON HYPER-PARAMETERS β

OVD-Explorer is sensitive to α, as shown in Section 5.4. There is another hyper-parameter β, which controls the scale of uncertainty quantification as shown in Equation 11 and Equation 13, further having an impact on $\bar{Z}^\pi$ and Z^π. To evaluate its sensitivity about β, we conduct the experiment

on Noisy Ant-v2 task using OVD-Explorer_G, and the result is shown in Figure 10. The results demonstrate that there is a broad range of settings for β, which can lead to well performance.

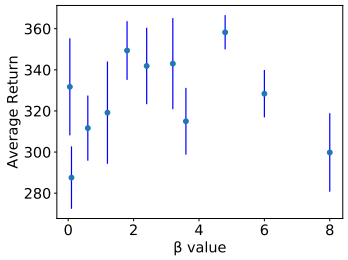
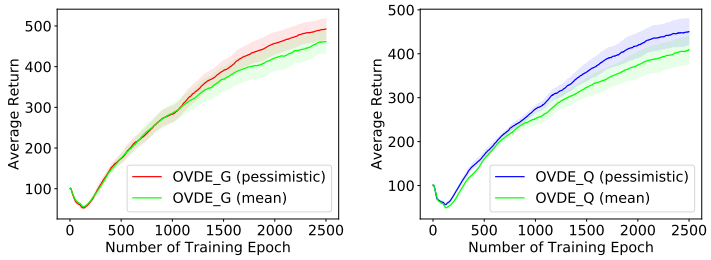


Figure 10: Sensitivity to β. The x-axis indicates different β settings, while the y-axis is the evaluation result represented by average episode return in the last epoch before totally 1250 epoch. Error bars indicate half standard deviation of average evaluation over 5 seeds. The 11 different β values are 0.05, 0.1, 0.6, 1.2, 1.8, 2.4, 3.2, 3.6, 4.8, 6.0, 8.0.

E.8 Ablation study on value distribution estimation

As mentioned in Section 3.2, we estimate Z^π pessimistically to alleviate over-estimation. Also, as mentioned in Appendix E.4, the pessimism is unnecessary if there is a need to explore areas with high aleatoric uncertainty. In the following, to investigate the benefit of pessimistic estimation in general case, we compare the performance of OVD-Explorer to the modified versions that use normally estimated Z^π. Our results show that pessimistic estimation can mostly be better than that using normal estimation.



(a) (b)

Figure 11: Training curves on Noisy Ant-v2 with different estimation of Z^π. The x-axis indicates number of training epoch (100 environment steps for each training epoch), while the y-axis is the evaluation result represented by average episode return. The shaded region denotes half standard deviation of average evaluation over 5 seeds. Curves are smoothed uniformly for visual clarity.

For OVDE_G (mean), we modified the mean of Gaussian distribution Z^π in Equation 13 from the lower bound to average value of the Q estimation $\mu(s, a)$ as in Equation 11. For OVDE_Q (mean), we modified the $z_i^\pi(s, a)$ in Equation 14 from the minimum value to the average value, i.e., $z_i^\pi(s, a; \theta) = \mathbb{E}_{k=1,2}\hat{Z}_i(s, a; \theta_k)$.

As shown in Figure 11, both OVDE_G (mean) and OVDE_Q (mean) perform worse than the pessimistic version. To draw a conclusion, pessimistic estimate is indeed required in general cases. Only when there is a need to explore areas with high aleatoric uncertainty, is such pessimistic estimation worth being turned off.

E.9 The performance compared with RND

RND also follows OFU principle, modeling uncertainty based on network distillation and using it as an intrinsic motivation signal to facilitate agent exploration. For the sake of fairness, we implement

RND based on DSAC, denoted by DSAC+RND in the following, and evaluate it on 3 standard Mujoco tasks and 3 noisy Mujoco tasks. We show the results in the following.

Table 8: Comparisons with RND. We report the averaged performance and standard deviation of 5 runs. Each trail uses the mean undiscounted return over the last 100 epoch. The maximum value of each row is shown in bold.

TASK	EPOCH	DSAC	DSAC+RND	OVD-EXPLORER_G	OVD-EXPLORER_Q
ANT-V2	2500	6206.9±1202.5	7308.4±641.3	7160.6±763.2	**7590.3**±154.9
HALFCHEETAH-V2	2500	13890.0±3424.4	12198.1±2338.3	14084.5±1579.8	**14792.4**±997.4
HOPPER-V2	1250	2199.7±602.7	2077.9±344.1	2239.5±428.2	**2619.3**±457.0
N-HALFCHEETAH-V2	1250	431.39±35.68	409.48±45.88	**445.28**±37.52	429.63±34.45
N-HOPPER-V2	1250	244.53±4.71	231.46±9.94	**252.09**±7.82	237.68±13.11
N-ANT-V2 (250)	1250	1275.87±172.64	1306.05±223.18	-	**1384.43**±84.39

For the complex task Ant-v2, RND brings much greater improvement by facilitating exploration based on the DSAC. For the other simpler tasks, RND does not bring significant performance improvement. This experiment demonstrates to some extent the effectiveness of RND exploration on several Mujoco tasks, but at the same time, our algorithm OVD-Explorer is still better.

E.10 ANALYSIS ABOUT EXPLORATION PROCESS OF OVD-EXPLORER

In the following, we further verify from statistical analysis that OVD-Explorer is in compliance with our raised exploration principle, i.e., **OVD-Explorer can achieve better trade-off between optimistic exploration and effectively avoiding exploring the areas with high aleatoric uncertainty**. We show the values of uncertainty estimations and our exploration objective (mutual information) at different stages during the training processes of two trials with different noise settings in figures.

In Figure 12, the environment noise is set lower around the goal, noting that the darker background color in the map represents higher aleatoric uncertainty, and the red dot represents the coordinate of the current state. The performance under this trail is given and the agent hardly ever reaches the goal before the 1000th epoch. Therefore, in the early stage, the aleatoric uncertainty is inaccurate and remains very low, as the value distribution shows little divergence. The figure also shows that our exploration objective (in green) is high when the epistemic uncertainty is high. So before the 1000th epoch, the exploration is guided by epistemic uncertainty, which follows the OFU principle. Later, once the goal has been explored, the aleatoric uncertainty is properly modelled, i.e., the aleatoric uncertainty towards left is larger than right at current state (see epoch 1240). Then the mutual information value towards left is lower than right, although the epistemic towards left is higher. It indicates that OVD-Explorer can property balance epistemic uncertainty and aleatoric uncertainty, and effectively avoid to explore the areas with higher aleatoric uncertainty.

In Figure 13, the environment noise is set higher around the goal. At the stage before the 1000th epoch, the exploration guidance is similar to Figure 12. The agent would hardly estimate the accurate aleatoric uncertainty without obtaining any reward. In the later stage, at the 1240th epoch, OVD-Explorer suggests exploring to the right, even though it has been recognized that the environmental uncertainty on the right is high. This is because the epistemic unertainty dominates under the mutual information. In contrast, at the 1249th epoch, when the action towards right has been explored much, the significant higher aleatoric uncertainty towards right dominates. Therefore, following the mutual information, the action towards left is preferred. This demonstrates the trade off that OVD-Explorer make, which satisfies our raised exploration principle.

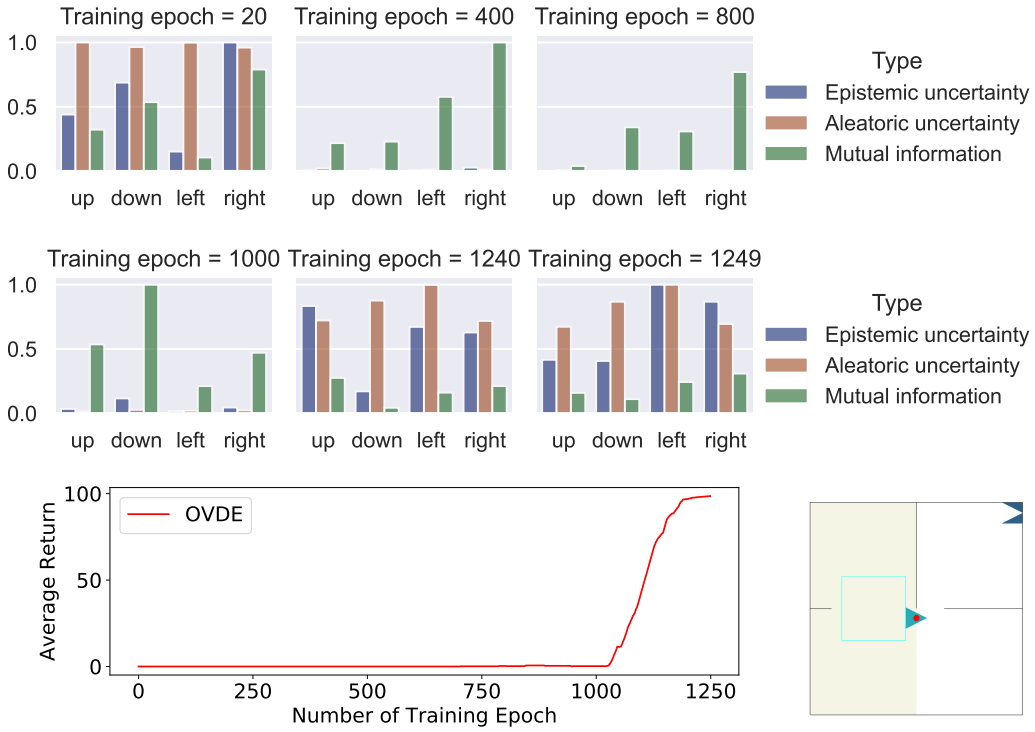


Figure 12: The statistical analysis for the training process, with the aleatoric uncertainty around the goal is set lower.

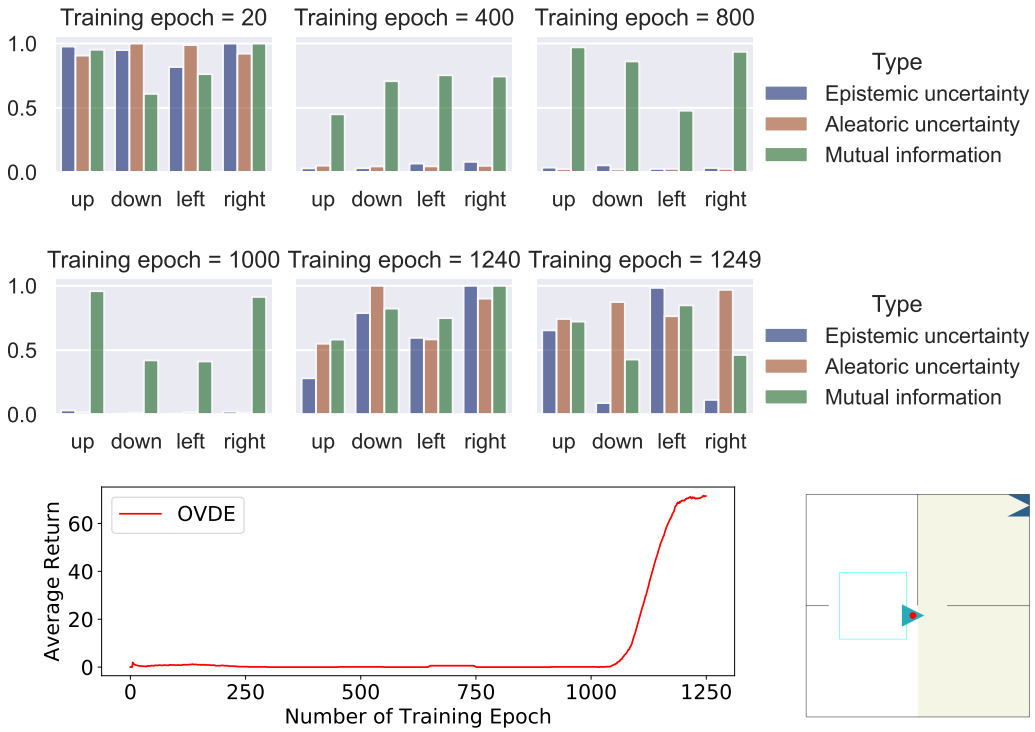


Figure 13: The statistical analysis for the training process, with the aleatoric uncertainty around the goal is set higher.