A Proofs

We first redefine notation for clarity and then provide the proofs of the results in the main paper.

Notation. Let $k \in N$ denote an iteration of policy evaluation(in Section 3.2). V^k denotes the true, tabular (or functional) V-function iterate in the MDP, without any correction. $\hat{V}^k$ denotes the approximate tabular (or functional) V-function iterate.

The empirical Bellman operator can be expressed as follows:

$$(\hat{\mathcal{B}}^\pi \hat{V}^k)(s) = E_{a\sim\pi(a|s)}\hat{r}(s,a) + \gamma \sum_{s'} E_{a\sim\pi(a|s)}\hat{P}(s'|s,a)[\hat{V}^k(s')] \tag{10}$$

where $\hat{r}(s,a)$ is the empirical average reward obtained in the dataset when performing action a at state s . The true Bellman operator can be expressed as follows:

$$(\mathcal{B}^\pi V^k)(s) = E_{a\sim\pi(a|s)}r(s,a) + \gamma \sum_{s'} E_{a\sim\pi(a|s)}P(s'|s,a)[V^k(s')] \tag{11}$$

Now we first prove that the iteration in Eq.2 has a fixed point. Assume state value function is lower bounded, i.e., $V(s) \geq C \; \forall s \in S$, then Eq.2 can always be solved with Eq.3. Thus, we only need to investigate the iteration in Eq.3.

Denote the iteration as a function operator $\mathcal{T}^\pi$ on V. Suppose $\operatorname{supp} d \subseteq \operatorname{supp} d_u$, then the operator $\mathcal{T}^\pi$ is a γ-contraction in L_∞ norm where γ is the discounting factor.

Proof of Lemma 3.1: Let V and V' are any two state value functions with the same support, i.e., $\mathtt{supp}V = \mathtt{supp}V'$.

$$\begin{aligned}
|(\mathcal{T}^\pi V - \mathcal{T}^\pi V')(s)| &= \left|(\hat{\mathcal{B}}^\pi V(s) - \alpha[\frac{d_{(s)}}{d_u(s)} - 1]) - (\hat{\mathcal{B}}^\pi V'(s) - \alpha[\frac{d_{(s)}}{d_u(s)} - 1])\right| \\
&= \left|\hat{\mathcal{B}}^\pi V(s) - \hat{\mathcal{B}}^\pi V'(s)\right| \\
&=|(E_{a\sim\pi(a|s)}\hat{r}(s,a) + \gamma E_{a\sim\pi(a|s)}\sum_{s'}\hat{P}(s'|s,a)V(s')) \\
&\quad - (E_{a\sim\pi(a|s)}\hat{r}(s,a) + \gamma E_{a\sim\pi(a|s)}\sum_{s'}\hat{P}(s'|s,a)V'(s'))| \\
&=\gamma\left|E_{a\sim\pi(a|s)}\sum_{s'}\hat{P}(s'|s,a)[V(s') - V'(s')]\right| \\
||\mathcal{T}^\pi V - \mathcal{T}^\pi V'||_\infty &= \max_s |(\mathcal{T}^\pi V - \mathcal{T}^\pi V')(s)| \\
&= \max_s \gamma \left|E_{a\sim\pi(a|s)}\sum_{s'}\hat{P}(s'|s,a)[V(s') - V'(s')]\right| \\
&\leq \gamma E_{a\sim\pi(a|s)}\sum_{s'}\hat{P}(s'|s,a)\max_{s''}|V(s'') - V'(s'')| \\
&= \gamma \max_{s''}|V(s'') - V'(s'')| \\
&= \gamma||(V - V')||_\infty
\end{aligned}$$

□

We present the bound on using empirical Bellman operator compared to the true Bellman operator. Following previous work [4], we make the following assumptions that: P^π is the transition matrix coupled with policy, specifically, $P^\pi V(s) = E_{a'\sim\pi(a'|s'),s'\sim P(s'|s,a')}[V(s')]$

Assumption A.1. $\forall s, a \in \mathcal{M}$, the following relationships hold with at least $(1-\delta)$ $(\delta \in (0,1))$ probability,

$$|r - r(s,a)| \leq \frac{C_{r,\delta}}{\sqrt{|D(s,a)|}}, ||\hat{P}(s'|s,a) - P(s'|s,a)||_1 \leq \frac{C_{P,\delta}}{\sqrt{|D(s,a)|}} \tag{12}$$

Under this assumption, the absolute difference between the empirical Bellman operator and the actual one can be calculated as follows:

$$
\begin{aligned}
|(\hat{\mathcal{B}}^\pi)\hat{V}^k - (\mathcal{B}^\pi)\hat{V}^k)| &= E_{a\sim\pi(a|s)}|r - r(s,a) + \gamma \sum_{s'} E_{a'\sim\pi(a'|s')}(\hat{P}(s'|s,a) - P(s'|s,a))[\hat{V}^k(s')]| && (13)\\
&\leq E_{a\sim\pi(a|s)}|r - r(s,a)| + \gamma |\sum_{s'} E_{a'\sim\pi(a'|s')}(\hat{P}(s'|s,a') - P(s'|s,a'))[\hat{V}^k(s')]| && (14)\\
&\leq E_{a\sim\pi(a|s)} \frac{C_{r,\delta} + \gamma C_{P,\delta} 2R_{max}/(1-\gamma)}{\sqrt{|D(s,a)|}} && (15)
\end{aligned}
$$

Thus, the estimation error due to sampling error can be bounded by a constant as a function of $C_{r,\delta}$ and $C_{t,\delta}$. We define this constant as $C_{r,T,\delta}$.

Thus we obtain:

$$
\forall V, s \in D, |\hat{\mathcal{B}}^\pi V(s) - \mathcal{B}^\pi V(s)| \leq E_{a\sim\pi(a|s)} \frac{C_{r,t,\delta}}{(1-\gamma)\sqrt{|D(s,a)|}} \tag{16}
$$

Next we provide an important lemma.

Lemma A.2. *(Interpolation Lemma) For any $f \in [0,1]$, and any given distribution $\rho(s)$, let d_f be an f-interpolation of ρ and D, i.e.,$d_f(s) := f d(s) + (1-f)\rho(s)$, let $v(\rho, f) := E_{s\sim\rho(s)}[\frac{\rho(s)-d(s)}{d_f(s)}]$, then $v(\rho, f)$ satisfies $v(\rho, f) \geq 0$.*

The proof can be found in [6]. By setting f as 1, we have $E_{s\sim\rho(s)}[\frac{\rho(s)-d(s)}{d(s)}] > 0$.

Proof of Theorem 3.2: The V function of approximate dynamic programming in iteration k can be obtained as:

$$
\hat{V}^{k+1}(s) = \hat{\mathcal{B}}^\pi \hat{V}^k(s) - \alpha[\frac{d(s)}{d_u(s)} - 1]\ \forall s, k \tag{17}
$$

The fixed point:

$$
\hat{V}^\pi(s) = \hat{\mathcal{B}}^\pi \hat{V}^\pi(s) - \alpha[\frac{d(s)}{d_u(s)} - 1] \leq \mathcal{B}^\pi \hat{V}^\pi(s) + E_{a\sim\pi(a|s)} \frac{C_{r,t,\delta} R_{max}}{(1-\gamma)\sqrt{|D(s,a)|}} - \alpha[\frac{d(s)}{d_u(s)} - 1] \tag{18}
$$

Thus we obtain:

$$
\hat{V}^\pi(s) \leq V^\pi(s) + (I - \gamma P^\pi)^{-1} E_{a\sim\pi(a|s)} \frac{C_{r,t,\delta} R_{max}}{(1-\gamma)\sqrt{|D(s,a)|}} - \alpha (I - \gamma P^\pi)^{-1}[\frac{d(s)}{d_u(s)} - 1] \tag{19}
$$

, where P^π is the transition matrix coupled with the policy π and $P^\pi V(s) = E_{a'\sim\pi(a'|s')s'\sim P(s'|s,a')}[V(s')]$.

Then the expectation of $V^\pi(s)$ under distribution $d(s)$ satisfies:

$$
\begin{aligned}
E_{s\sim d(s)} \hat{V}^\pi(s) \leq & E_{s\sim d(s)}(V^\pi(s)) + E_{s\sim d(s)}(I - \gamma P^\pi)^{-1} E_{a\sim\pi(a|s)} \frac{C_{r,t,\delta} R_{max}}{(1-\gamma)\sqrt{|D(s,a)|}} \\
& -\alpha \underbrace{E_{s\sim d(s)}(I - \gamma P^\pi)^{-1}[\frac{d(s)}{d_u(s)} - 1])}_{>0}
\end{aligned} \tag{20}
$$

When $\alpha \geq \frac{E_{s\sim d(s)} E_{a\sim\pi(a|s)} \frac{C_{r,t,\delta} R_{max}}{(1-\gamma)\sqrt{|D(s,a)|}}}{E_{s\sim d(s)}[\frac{d(s)}{d_u(s)} - 1])}$, $E_{s\sim d(s)} \hat{V}^\pi(s) \leq E_{s\sim d(s)}(V^\pi(s))$. □

Proof of Theorem 3.3: The expectation of $V^\pi(s)$ under distribution $d(s)$ satisfies:

$$\begin{aligned} E_{s\sim d_u(s)}\hat{V}^\pi(s) \leq & E_{s\sim d_u(s)}(V^\pi(s)) + E_{s\sim d_u(s)}(I-\gamma P^\pi)^{-1} E_{a\sim\pi(a|s)} \frac{C_{r,t,\delta}R_{max}}{(1-\gamma)\sqrt{|D(s,a)|}} \\ & - \alpha E_{s\sim d_u(s)}(I-\gamma P^\pi)^{-1}[\frac{d(s)}{d_u(s)}-1]) \end{aligned} \tag{21}$$

Noticed that the last term:

$$\sum_{s\sim d_u(s)}(\frac{d_f(s)}{d_u(s)}-1) = \sum_s d_u(s)(\frac{d_f(s)}{d_u(s)}-1) = \sum_s d_f(s) - \sum_s d_u(s) = 0 \tag{22}$$

We obtain that:

$$E_{s\sim d_u(s)}\hat{V}^\pi(s) \leq E_{s\sim d_u(s)}(V^\pi(s)) + E_{s\sim d_u(s)}(I-\gamma P^\pi)^{-1} E_{a\sim\pi(a|s)} \frac{C_{r,t,\delta}R_{max}}{(1-\gamma)\sqrt{|D(s,a)|}} \tag{23}$$

□

Proof of Theorem 3.4: Recall that the expression of the V-function iterate is given by:

$$\hat{V}^{k+1}(s) = \mathcal{B}^{\pi^k}\hat{V}^k(s) - \alpha[\frac{d(s)}{d_u(s)}-1]\forall s,k \tag{24}$$

Now the expectation of $V^\pi(s)$ under distribution $d_u(s)$ is given by:

$$E_{s\sim d_u(s)}\hat{V}^{k+1}(s) = E_{s\sim d_u(s)}\left[\mathcal{B}^{\pi^k}\hat{V}^k(s) - \alpha[\frac{d(s)}{d_u(s)}-1]\right] = E_{s\sim d_u(s)}\mathcal{B}^{\pi^k}\hat{V}^k(s) \tag{25}$$

The expectation of $V^\pi(s)$ under distribution $d(s)$ is given by:

$$E_{s\sim d(s)}\hat{V}^{k+1}(s) = E_{s\sim d(s)}\mathcal{B}^{\pi^k}\hat{V}^k(s) - \alpha[\frac{d(s)}{d_u(s)}-1] = E_{s\sim d(s)}\mathcal{B}^{\pi^k}\hat{V}^k(s) - \alpha E_{s\sim d(s)}[\frac{d(s)}{d_u(s)}-1] \tag{26}$$

Thus we can show that:

$$\begin{aligned} E_{s\sim d_u(s)}\hat{V}^{k+1}(s) - E_{s\sim d(s)}\hat{V}^{k+1}(s) &= E_{s\sim d_u(s)}\mathcal{B}^{\pi^k}\hat{V}^k(s) - E_{s\sim d(s)}\mathcal{B}^{\pi^k}\hat{V}^k(s) + \alpha E_{s\sim d(s)}[\frac{d(s)}{d_u(s)}-1] \\ &= E_{s\sim d_u(s)}V^{k+1}(s) - E_{s\sim d(s)}V^{k+1}(s) - E_{s\sim d(s)}[\mathcal{B}^{\pi^k}(\hat{V}^k - V^k)] \\ &+ E_{s\sim d_u(s)}[\mathcal{B}^{\pi^k}(\hat{V}^k - V^k)] + \alpha E_{s\sim d(s)}[\frac{d(s)}{d_u(s)}-1] \end{aligned} \tag{27}$$

By choosing α:

$$\alpha > \frac{E_{s\sim d(s)}[\mathcal{B}^{\pi^k}(\hat{V}^k - V^k)] - E_{s\sim d_u(s)}[\mathcal{B}^{\pi^k}(\hat{V}^k - V^k)]}{E_{s\sim d(s)}[\frac{d(s)}{d_u(s)}-1]} \tag{28}$$

We have $E_{s\sim d_u(s)}\hat{V}^{k+1}(s) - E_{s\sim d(s)}\hat{V}^{k+1}(s) > E_{s\sim d_u(s)}V^{k+1}(s) - E_{s\sim d(s)}V^{k+1}(s)$ hold. □

Proof of Theorem 3.5: $\hat{V}$ is obtained by solving the recursive Bellman fixed point equation in the empirical MDP, with an altered reward, $r(s,a) - \alpha[\frac{d(s)}{d_u(s)}-1]$, hence the optimal policy $\pi^*(a|s)$ obtained by optimizing the value under Eq. 3.5. □

Proof of Theorem 3.6: The proof of this statement is divided into two parts. We first relates the return of π^* in the empirical MDP $\hat{M}$ with the return of π_β , we can get:

$$J(\pi^*,\hat{M}) - \alpha\frac{1}{1-\gamma}\mathbb{E}_{s\sim d^{\pi^*}_{\hat{M}}(s)}[\frac{d(s)}{d_u(s)}-1] \geq J(\pi_\beta,\hat{M}) - 0 = J(\pi_\beta,\hat{M}) \tag{29}$$

The next step is to bound the difference between $J(\pi_\beta,\hat{M})$ and $J(\pi_\beta,M)$ and the difference between $J(\pi^*,\hat{M})$ and $J(\pi^*,M)$. We quote a useful lemma from [4] (Lemma D.4.1):

Algorithm 1 CSVE based Offline RL Algorithm

Input: data $D = \{(s, a, r, s')\}$
Parametered Models: Q_θ, V_ψ, π_ϕ, $Q_{\overline{\theta}}$, M_ν
Hyperparameters: α, λ, learning rates $\eta_\theta, \eta_\psi, \eta_\phi, \omega$
▷ *Train the transition model with the static dataset D*
$M_\nu \leftarrow \mathit{train}(D)$.
▷ *Train the conservative value estimation and policy functions*
Initialize function parameters $\theta_0, \psi_0, \phi_0, \overline{\theta}_0 = \theta_0$
for step $k = 1$ **to** N **do**
  $\psi_k \leftarrow \psi_{k-1} - \eta_\psi \nabla_\psi L_V^\pi(V_\psi; \overline{\hat{Q}_{\theta_k}})$
  $\theta_k \leftarrow \theta_{k-1} - \eta_\theta \nabla_\theta L_Q^\pi(Q_\theta; \hat{V}_{\psi_k})$
  $\phi_k \leftarrow \phi_{k-1} - \eta_\phi \nabla_\phi L_\pi^+(\pi_\phi)$
  $\overline{\theta}_k \leftarrow \omega \overline{\theta}_{k-1} + (1-\omega)\theta_k$
end for

Lemma A.3. *For any MDP M, an empirical MDP $\hat{M}$ generated by sampling actions according to the behavior policy π_β and a given policy π,*

$$|J(\pi, \hat{M}) - J(\pi, M)| \leq (\frac{C_{r,\delta}}{1-\gamma} + \frac{\gamma R_{max} C_{T,\delta}}{(1-\gamma)^2}) \mathbb{E}_{s \sim d_{\hat{M}}^{\pi^*}(s)} [\frac{\sqrt{|\mathcal{A}|}}{\sqrt{|\mathcal{D}(s)|}} \sqrt{E_{a \sim \pi(a|s)} (\frac{\pi(a|s)}{\pi_\beta(a|s)})}] \quad (30)$$

Setting π in the above lemma as π_β, we get:

$$|J(\pi_\beta, \hat{M}) - J(\pi_\beta, M)| \leq (\frac{C_{r,\delta}}{1-\gamma} + \frac{\gamma R_{max} C_{T,\delta}}{(1-\gamma)^2}) \mathbb{E}_{s \sim d_{\hat{M}}^{\pi^*}(s)} [\frac{\sqrt{|\mathcal{A}|}}{\sqrt{|\mathcal{D}(s)|}} \sqrt{E_{a \sim \pi^*(a|s)} (\frac{\pi^*(a|s)}{\pi_\beta(a|s)})}] \quad (31)$$

given that $\sqrt{E_{a\sim\pi^*(a|s)}[\frac{\pi^*(a|s)}{\pi_\beta(a|s)}]}$ is a pointwise upper bound of $\sqrt{E_{a\sim\pi_\beta(a|s)}[\frac{\pi_\beta(a|s)}{\pi_\beta(a|s)}]}$([4]). Thus we get,

$$\begin{aligned} J(\pi^*, \hat{M}) \geq J(\pi_\beta, \hat{M}) - 2(\frac{C_{r,\delta}}{1-\gamma} + \frac{\gamma R_{max} C_{T,\delta}}{(1-\gamma)^2}) \mathbb{E}_{s \sim d_{\hat{M}}^{\pi^*}(s)} [\frac{\sqrt{|\mathcal{A}|}}{\sqrt{|\mathcal{D}(s)|}} \sqrt{E_{a \sim \pi^*(a|s)} (\frac{\pi^*(a|s)}{\pi_\beta(a|s)})}] \\ + \alpha \frac{1}{1-\gamma} \mathbb{E}_{s \sim d_{\hat{M}}^{\pi}(s)} [\frac{d(s)}{d_u(s)} - 1] \end{aligned} \quad (32)$$

which completes the proof. □

Here, the second term is sampling error which occurs due to mismatch of $\hat{M}$ and M; the third term denotes the increase in policy performance due to CSVE in $\hat{M}$. Note that when the first term is small, the smaller value of α are able to provide an improvement compared to the behavior policy.

B CSVE Algorithm and Implementation Details

In section 4, we have given the complete formula descriptions of a practical offline RL algorithm of CSVE. Here we put all together and describe the practical deep offline reinforcement learning algorithm in Alg. 1. In particular, the dynamic model model, value functions and policy are all parameterized with deep neural networks and trained via stochastic gradient decent methods.

We implement our method based on an offline deep reinforcement learning library d3rlpy [33]. The code is available at: https://github.com/2023AnnonymousAuthor/csve .

B.1 Additional ablation study

Effect of exploration on near states. We analyze the impact of varying the factor λ in Eq. 9, which controls the intensity on such exploration. We investigated λ values of $\{0.0, 0.1, 0.5, 1.0\}$ in the

Table 3: Hyper-parameters of CSVE evaluation

Hyper-parameters	Value and description
B	5, number of ensembles in dynamics model
α	10, to control the penalty of out-of-distribution states
τ	10, budget parameter in Eq. 8
β	In Gym domain, 3 for random and medium tasks, 0.1 for the other tasks; In Adroit domain, 30 for human and cloned tasks, 0.01 for expert tasks
γ	0.99, discount factor.
H	1 million for Mujoco while 0.1 million for Adroit tasks.
w	0.005, target network smoothing coefficient.
lr of actor	3e-4, policy learning rate
lr of critic	1e-4, critic learning rate

medium tasks, fixing $\beta = 0.1$. The results are plotted in Fig. 2. As shown in the upper figures, λ has obvious effect to policy performance and variances during training. With increasing λ from 0, the converged performance gets better in general. However, when the λ becomes too large (e.g., $\lambda = 3$ for hopper and walker2d), the performance may degrade or even collapse. By further investigating the L_π loss in Eq.9, as shown in the bottom figures, we found that larger λ values have negative effect to L_π; however, once L_π converges low reasonably, the bigger λ leads to performance improvement.

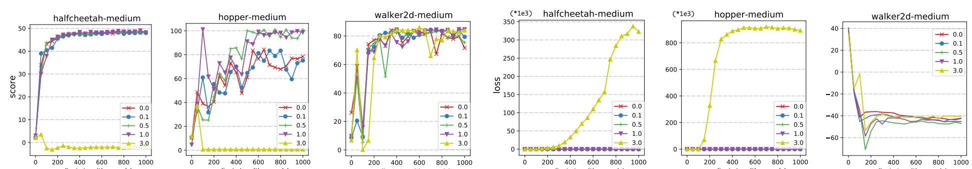

Figure 2: Effect of λ to performance scores (upper figures) and losses (bottom figures) in Eq. 9 on medium tasks.

Effect of model errors. Compared to traditional model-based offline RL algorithms, CSVE is less affected by model biases. To measure this quantitatively, we studied the impact of model biases on performance by using the average L2 error on transition prediction as a surrogate for model biases. As shown in Fig. 3, the effect of model bias on RL performance is subtle in CSVE. Specifically, for the halfcheetah task, there is no observable impact of model errors on scores, while in the hopper and walker2d tasks, there is only a slight decrease in scores as the errors increase.

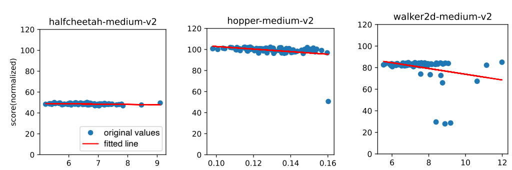

Figure 3: Effect of the model biases to performance scores. The correlation coefficient is -0.32, -0.34, and -0.29 respectively.

C Experimental Details and Complementary Results

C.1 Hyper-parameters of CSVE evaluation in experiments

The detailed hyper-parameters of CSVE used in experiments are provided in Table 3.

C.2 More experiments on hyper-parameters effect

We also investigated λ values of $\{0.0, 0.5, 1.0, 3.0\}$ in the medium-replay tasks. The results are shown in Fig. 2.

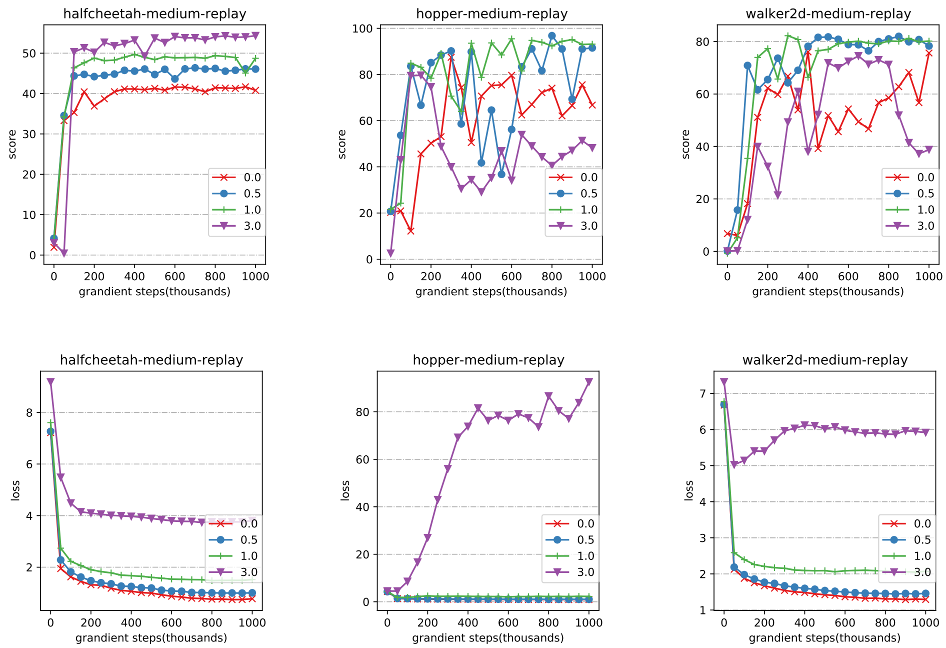

Figure 4: Effect of λ to Score (upper figures) and L_π loss in Eq. 9 (bottom figures)

C.3 Details of Baseline CQL-AWR

In order to directly compare effect of the conservative state value estimation against Q value estimation, we implement a baseline method namely CQL-AWR as follows:

$$
\begin{aligned}
\hat{Q}^{k+1} &\leftarrow \arg\min_{Q} \alpha \left(E_{s\sim D, a\sim\pi(a|s)}[Q(s,a)] - E_{s\sim D, a\sim\hat{\pi}_\beta(a|s)}[Q(s,a)] \right) + \frac{1}{2} E_{s,a,s'\sim D}[(Q(s,a) - \hat{\beta}_\pi \hat{Q}^k(s,a))^2] \\
\pi &\leftarrow \arg\min_{\pi'} L_\pi(\pi') = -E_{s,a\sim D}\left[\log \pi'(a|s) \exp\left(\beta \hat{A}^{k+1}(s,a)\right)\right] - \lambda E_{s\sim D, a\sim\pi'(s)}\left[\hat{Q}^{k+1}(s,a)\right] \\
&\quad \text{where } \hat{A}^{k+1}(s,a) = \hat{Q}^{k+1}(s,a) - \mathbb{E}_{a\sim\pi}[\hat{Q}^{k+1}(s,a)].
\end{aligned}
$$

In CQL-AWR, the critic adopts a normal CQL equation, while the policy improvement part uses a AWR extended with new action exploration indicated by the conservative Q function. Compared to our CSVE implementation, its policy part is similar except that the exploration is Q-based and model-free.

C.4 Reproduction of COMBO

In Table 1 of our main paper, our results of COMBO adopt the one presented in literature [23]. Here we list other reproducing efforts and results which may be helpful for readers to compare CSVE with COMBO.

Official Code. We preferred to rerun the official COMBO code provided by authors. The code is implemented in Tensoflow 1.x and depends on software versions in 2018. We rebuilt the environment but still failed to reproduce the results. For example, Fig. 5 shows the asymptotic performance on medium datasets until 1000 epochs, in which the scores have been normalized with corresponding SAC performance. We found that in both hopper and walker2d, the scores show dramatic fluctuations. The average scores of last 10 epochs for halfcheetah, hopper and walker2d are 71.7, 65.3 and -0.26 in respect. Besides, we found that even in D4RL v0 dataset, COMBO's behaviours are similar with recommended hyper-parameters.

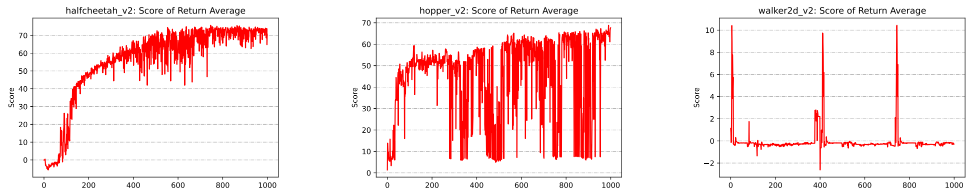

Figure 5: Return of official COMBO implementation on D4RL mujoco v2 tasks, fixing seed=0.

JAX-based optimized implementation Code [34]. We also rerun one recent re-implementation in RIQL which is the most highly tuned implementation so far. The results are shown in Fig.6. For random and expert datasets, we used the same hyper-parameters same with medium and medium-expert respectively. For all other datasets, we used the default hyper-parameters given by authors [34]. By comparing with the authors' results (Table 10 and Fig.7 in [34]), our reproduced results are still lower and with larger variances.

C.5 Effect of Exploration on near Dataset Distributions

As discussed in Section 3.1 and 4.2, proper choice of exploration on the distribution (d) beyond data (d_u) should help policy improvement. The factor λ in Eq. 9 controls the trade-off on such 'bonus' exploration and complying the data-implied behaviour policy.

In section 5.2, we have investigated the effect of λ in medium datasets of mujoco tasks. Now let us further take the medium-replay type of datasets for more analysis of its effect. In the experiments, with fixed $\beta = 0.1$, we investigate λ values of $\{0.0, 0.5, 1.0, 3.0\}$. As shown in the upper figures in Fig. 4, λ shows obvious effect to policy performance and variances during training. In general, there is a value under which increasing λ leads to performance improvement, while above which further increasing λ hurts performance. For example, with $\lambda = 3.0$ in hopper-medium-replay task and walker2d-medium-replay task, the performance get worse than with smaller λ values. The value of λ is task-specific, and we find that its effect is highly related to the loss in Eq. 9 which can be observed by comparing bottom and upper figures in Fig. 4. Thus, in practice, we can choose proper λ according to the above loss without online interaction.

C.6 Conservative State Value Estimation by Perturbing Data State with Noise

In this section, we investigate a model-free method for sampling OOD states, and compare its results with the model-based method adopted in our implementation in section 4.

The model-free method samples OOD states by randomly adding Gaussian noise to the sampled states from data. Specifically, we replace the Eq.5 with the following Eq. 33, and other parts are same as previous.

$$\begin{aligned}\hat{V}^{k+1} \leftarrow \arg\min_{V} L_V^{\pi}(V; \overline{\hat{Q}^k}) = \ & \alpha \left(E_{s\sim D, s'=s+N(0,\sigma^2)}[V(s')] - E_{s\sim D}[V(s)] \right) \\ & + E_{s\sim D}\left[\left(E_{a\sim\pi(\cdot|s)}[\overline{\hat{Q}^k}(s,a)] - V(s) \right)^2 \right].\end{aligned} \tag{33}$$

The experimental results on mujoco control tasks are summarized in Table 4. As shown, with different noise levels (σ^2), the model-free CSVE may performs better or worse than our original model-based CSVE implementation; and for some problems, the model-free method show very large variances across seeds. Intuitively, if the noise level covers the reasonable state distribution around data, its performance is good; otherwise, it misbehaves. Unfortunately, it is hard to find a noise level that is consistent for different tasks or even the same tasks with different seeds.

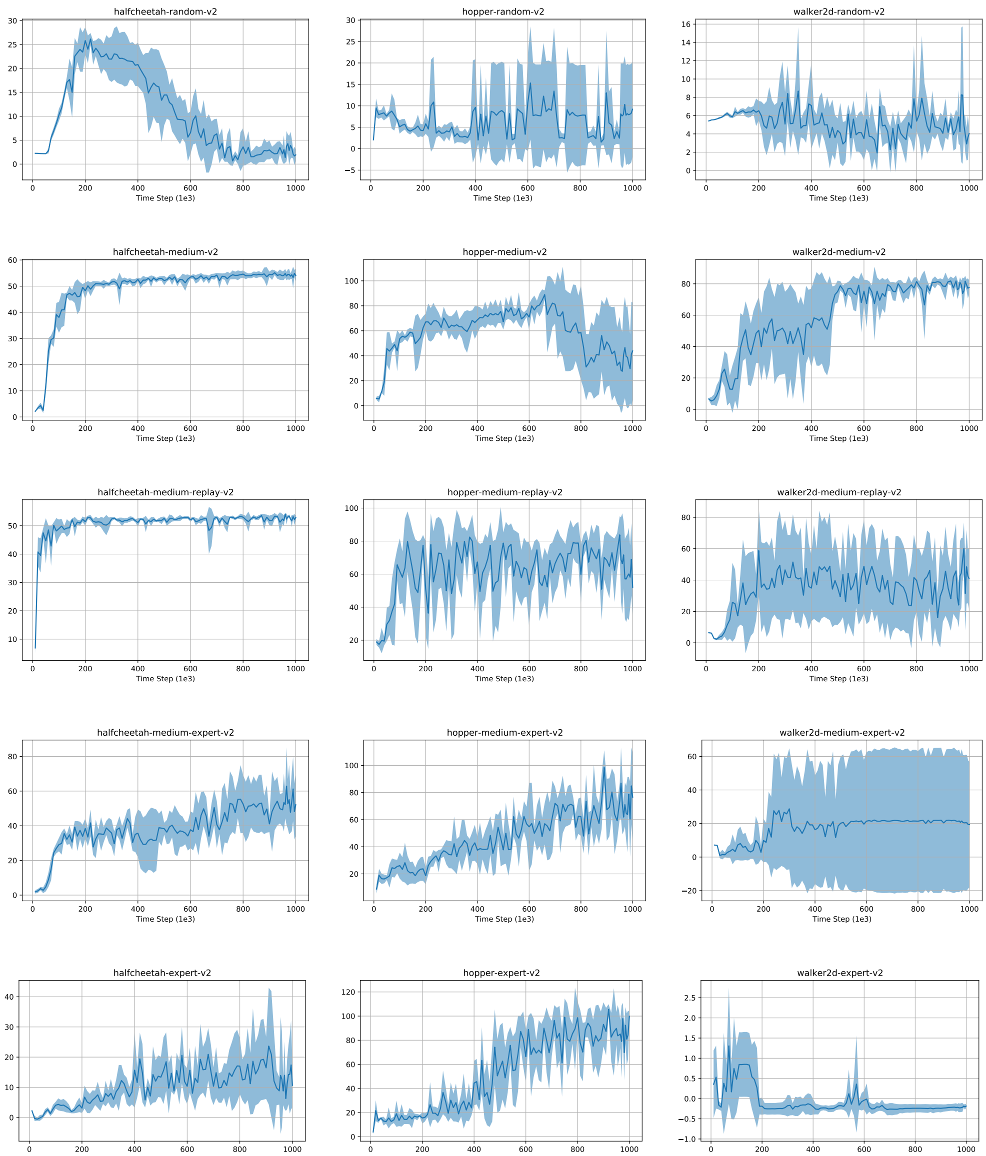

Figure 6: Return of an optimized COMBO implementation[34] on D4RL mujoco v2 tasks. The data are got by running with 5 seeds for each task, and the dynamics model has 7 ensembles.

Table 4: Performance comparison on Gym control tasks. The results of different noise levels is over three seeds.

		CQL	CSVE	σ^2=0.05	σ^2=0.1	σ^2=0.15
Random	HalfCheetah	17.5 ± 1.5	26.7 ± 2.0	20.8 ± 0.4	20.4 ± 1.3	18.6 ± 1.1
	Hopper	7.9 ± 0.4	27.0 ± 8.5	4.5 ± 3.1	14.2 ± 15.3	6.7 ± 5.4
	Walker2D	5.1 ± 1.3	6.1 ± 0.8	3.9 ± 3.8	7.5 ± 6.9	1.7 ± 3.5
Medium	HalfCheetah	47.0 ± 0.5	48.6 ± 0.0	48.2 ± 0.2	47.5 ± 0.0	46.0 ± 0.9
	Hopper	53.0 ± 28.5	99.4 ± 5.3	36.9 ± 32.6	46.1 ± 2.1	18.4 ± 30.6
	Walker2D	73.3 ± 17.7	82.5 ± 1.5	81.5 ± 1.0	75.5 ± 1.9	$78.6 \pm 2,9$
Medium Replay	HalfCheetah	45.5 ± 0.7	54.8 ± 0.8	44.8 ± 0.4	44.1 ± 0.5	43.8 ± 0.4
	Hopper	88.7 ± 12.9	91.7 ± 0.3	85.5 ± 3.0	78.3 ± 4.3	70.2 ± 12.0
	Walker2D	81.8 ± 2.7	78.5 ± 1.8	78.7 ± 3.3	76.8 ± 1.3	66.8 ± 4.0
Medium Expert	HalfCheetah	75.6 ± 25.7	93.1 ± 0.3	87.5 ± 6.0	89.7 ± 6.6	93.8 ± 1.6
	Hopper	105.6 ± 12.9	95.2 ± 3.8	63.2 ± 54.4	99.0 ± 11.0	37.6 ± 63.9
	Walker2D	107.9 ± 1.6	109.0 ± 0.1	108.4 ± 1.9	109.5 ± 1.3	110.4 ± 0.6
Expert	HalfCheetah	96.3 ± 1.3	93.8 ± 0.1	59.0 ± 28.6	67.5 ± 21.9	75.3 ± 27.3
	Hopper	96.5 ± 28.0	111.2 ± 0.6	67.3 ± 57.7	109.2 ± 2.4	109.4 ± 2.1
	Walker2D	108.5 ± 0.5	108.5 ± 0.0	109.7 ± 1.1	108.9 ± 1.6	108.6 ± 0.3