This document serves as the supplementary material to the AAAI 2025 paper ”Epistemic Bellman Operators”.

A Proofs

Proof of Theorem 1. Finding a method to utilize the contraction properties of the inner Bellman operator is the main challenge in this proof. We will achieve this by defining an Epistemic Bellman operator on the joint space of two sets of Q-values with correlated noise. This will cause the noise to cancel later in the proof and allows us to use the inner Bellman operator as contraction.

Recall that the Wasserstein distance can be written as

$$W_p(P_X, P_Y) = \left(\inf_{R \in \mathcal{R}_{XY}} \mathbb{E}_{(X,Y)\sim R}(\|X - Y\|^p) \right)^{\frac{1}{p}},$$

where $\mathcal{R}_{XY}$ is the set of joint probability measures that has marginals P_X and P_Y, and $\| \cdot \|$ is the norm on $\mathbb{R}^{|S||A|}$ in which $B_\mathcal{D}$ is assumed to be a γ-contraction.

We can define an operator analogous to the EBO that works on the joint space instead:

$$\hat{\mathcal{B}}R(X, Y) = \text{Law}\left((B_\mathcal{D}(X) + \epsilon_\mathcal{D},\ B_\mathcal{D}(Y) + \epsilon_\mathcal{D}, (X, Y) \sim R(X, Y)\right),$$

where crucially both $\epsilon_\mathcal{D}$ are the **same** variable. We are allowed to choose $\epsilon_\mathcal{D}$ in this manner because it leaves the marginal distributions unchanged.

Because $\hat{\mathcal{B}}\bar{R}$ has marginals $\mathcal{B}P_X$ and $\mathcal{B}P_Y$ if $\bar{R}$ has marginals P_X and P_Y, we have

$$W_p(\mathcal{B}P_X, \mathcal{B}P_Y)^p = \inf_{R \in \mathcal{R}_{\mathcal{B}P_X, \mathcal{B}P_Y}} \mathbb{E}_R\left[\|X - Y\|^p\right] \le \inf_{\bar{R} \in \mathcal{R}_{P_X, P_Y}} \mathbb{E}_{\hat{\mathcal{B}}\bar{R}}\left[\|X - Y\|^p\right]. \tag{1}$$

The inequality holds because the first infimum is over all R with marginals $\mathcal{B}P_X$ and $\mathcal{B}P_Y$, while the second infimum is restricted to the image of $\hat{\mathcal{B}}$.

Finally, we can conclude that

$$\begin{aligned} W_p(\mathcal{B}P_X, \mathcal{B}P_Y)^p &\le \inf_{\bar{R} \in \mathcal{R}_{P_X, P_Y}} \mathbb{E}_{\hat{\mathcal{B}}\bar{R}}\left[\|X - Y\|^p\right] = \inf_{\bar{R}} E_{\bar{R}} \|B_\mathcal{D}(X) + \epsilon_\mathcal{D} - B_\mathcal{D}(Y) - \epsilon_\mathcal{D}\|^p && (2) \\ &= \inf_{\bar{R}} E_{\bar{R}} \|B_\mathcal{D}(X) - B_\mathcal{D}(Y)\|^p \le \inf_{\bar{R}} E_{\bar{R}} \gamma^p \|X - Y\|^p \le \gamma^p \inf_{\bar{R}} E_{\bar{R}} \|X - Y\|^p && (3) \\ &= (\gamma W_p(P_X, P_Y))^p, && (4) \end{aligned}$$

where in Equation 3 we use the fact that $B_\mathcal{D}$ is a γ-contraction in $\|.\|$. □

Proof of Theorem 2. Most of this proof relies on the fact that we can switch the order of expectations and affine functions.

Since P_B is a fixed point of $\mathcal{B}$, we have

$$\mathbb{E}_{P_B}[Q] = \mathbb{E}_{\mathcal{B}P_B}[Q] = \mathbb{E}_{P_\epsilon}\mathbb{E}_{P_B}[B(Q) + \epsilon] = \mathbb{E}_{P_B}[B(Q)] = B[\mathbb{E}_{P_B}[Q]], \tag{5}$$

where the second equality is the definition of the EBO, the third equality follows from independence of ϵ and Q, and the last equality uses the fact that B is an affine function, proving that $\mathbb{E}_{P_B}[Q]$ is a fixed point of B.

For the rest of the proof, we will write BQ as shorthand for $B(Q)$, and since B is affine we define A and b such that $BQ = AQ + b$.

For the covariance we have

$$\begin{aligned} \mathbb{E}_{P_B}\left[QQ^\top\right] &= \mathbb{E}_{P_\epsilon}\left[\mathbb{E}_{P_B}\left[(BQ + \epsilon)(BQ + \epsilon)^\top\right]\right] \\ &= \mathbb{E}_{P_\epsilon}\left[\mathbb{E}_{P_B}\left[(BQ)(BQ)^\top + BQ\epsilon^\top + \epsilon(BQ)^\top + \epsilon\epsilon^\top\right]\right] \\ &= \mathbb{E}_{P_B}\left[(BQ)(BQ)^\top\right] + \mathbb{E}_{P_\epsilon}\mathbb{E}_{P_B}\left[BQ\epsilon^\top\right] + \mathbb{E}_{P_\epsilon}\mathbb{E}_{P_B}\left[\epsilon(BQ)^\top\right] + \mathbb{E}_{P_\epsilon}\left[\epsilon\epsilon^\top\right] \\ &= \mathbb{E}_{P_B}\left[(BQ)(BQ)^\top\right] + \mathbb{E}_{P_\epsilon}\left[\epsilon^\top\right]\mathbb{E}_{P_B}\left[BQ\right] + \mathbb{E}_{P_\epsilon}\left[\epsilon\right]\mathbb{E}_{P_B}\left[(BQ)^\top\right] + \mathbb{E}_{P_\epsilon}\left[\epsilon\epsilon^\top\right] \\ &= \mathbb{E}_{P_B}\left[(BQ)(BQ)^\top\right] + \mathbb{E}_{P_\epsilon}\left[\epsilon\epsilon^\top\right] \\ &= \mathbb{E}_{P_B}\left[(BQ)(BQ)^\top\right] + \Sigma_\epsilon \\ &= \mathbb{E}_{P_B}\left[(AQ + b)(Q^\top A^\top + b^\top\right] + \Sigma_\epsilon \\ &= A\mathbb{E}_{P_B}\left[QQ^\top\right]A^\top + b\mathbb{E}_{P_B}\left[Q^\top\right] + \mathbb{E}_{P_B}\left[Q\right]b^\top + bb^\top + \Sigma_\epsilon \\ &= A\mathbb{E}_{P_B}\left[QQ^\top\right]A^\top + b\mathbb{E}_{P_B}\left[Q^\top\right] + \mathbb{E}_{P_B}\left[Q\right]b^\top + bb^\top + \Sigma_\epsilon \\ &= A\mathbb{E}_{P_B}\left[QQ^\top\right]A^\top + bQ_B^\top + Q_B b^\top + bb^\top + \Sigma_\epsilon, \end{aligned} \tag{6}$$

and

$$Q_B Q_B^\top = (BQ_B)(BQ_B)^\top = AQ_B Q_B^\top A^\top + bQ_B^\top + Q_B b^\top + bb^\top,$$

so

$$\begin{aligned} \mathbb{E}_{P_B}\left[QQ^\top - Q_B Q_B^\top\right] &= A\mathbb{E}_{P_B}\left[QQ^\top\right]A^\top - AQ_B Q_B^\top A^\top + \Sigma_\epsilon \\ &= A\mathbb{E}_{P_B}\left[QQ^\top - Q_B Q_B^\top\right]A^\top + \Sigma_\epsilon. \end{aligned} \tag{7}$$

Vectorizing both sides yields

$$\text{Vec}(\mathbb{E}_{P_B}\left[QQ^\top - Q_B Q_B^\top\right]) - \text{Vec}(A\mathbb{E}_{P_B}\left[QQ^\top - Q_B Q_B^\top\right]A^\top) = \tag{8}$$

$$\text{Vec}(\mathbb{E}_{P_B}\left[QQ^\top - Q_B Q_B^\top\right]) - (A \otimes A)\text{Vec}(\mathbb{E}_{P_B}\left[QQ^\top - Q_B Q_B^\top\right]) = \tag{9}$$

$$\text{Vec}(\Sigma_\epsilon). \tag{10}$$

Finally, since B is a contraction, the absolute eigenvalues of A must be strictly smaller than 1. By basic properties of the Kronecker product, $A \otimes A$ then also has absolute eigenvalues strictly smaller than one. Therefore, $I - (A \otimes A)$ has only non-zero eigenvalues, and hence is invertible.

We conclude that

$$\text{Vec}(\mathbb{E}_{P_B}\left[QQ^\top - Q_B Q_B^\top\right]) = (I - A \otimes A)^{-1}\text{Vec}(\Sigma_\epsilon)$$

to finish the proof. □

B Experiment Details

Epistemic Bellman Operator

The MDP in this experiment has reward function $R = [0.05192758, -0.7084503]$, and the evaluated policy is $\pi = [0.4352794, 0.5647206]$. We use a standard 1 step Bellman operator B such that

$$BQ = R + \gamma T^\pi Q,$$

and the likelihood is given by $p(Q|q') = Bq' + \epsilon$ where $\epsilon \sim \mathcal{N}(0, \texttt{diag}(0.2, 0.2))$. The initial distribution is a normal distribution with mean 0 and covariance $\texttt{diag}(2, 2)$.

Code that generated the figures is available at github.com/pascal314/epistemic-bellman-operators. These experiments were completed on a single CPU within a few minutes.

Tabular Thompson Sampling

The MDP in this experiment has 30 states and 5 actions, with $\gamma = 0.9$. The transition function is drawn form a uniform dirichlet distribution, and the reward function is sampled i.i.d. from a normal distribution with standard deviation 0.1. We provide the true transition and reward function to all algorithms, but inject normally distributed noise ϵ with varying standard deviation.

Our MCMC sampler is a basic Hamiltonian Monte Carlo (HMC) sampler applied to the likelihood

$$p(Q) \propto \exp(\sum_{s,a} \frac{1}{2\sigma^2}(Q - R - \gamma T^{\pi^*_Q} Q))^2,$$

where σ is the true standard deviation of the injected noise. The Double-Q MCMC sampler is applied to the likelihood

$$p(Q, Q') \propto \exp\left(\sum_{s,a} \frac{1}{2\sigma^2}(Q - R - \gamma T^{\pi^*_{Q'}} Q)^2 + \frac{1}{2\sigma^2}(Q' - R - \gamma T^{\pi^*_Q} Q')^2\right),$$

The HMC hyperparameters are hand-tuned to have an acceptance rate around 0.9.

Our EBO sampler samples from the fixed point of the EBO by initializing a table of Q-values and repeatedly applying the EBO. The Double-Q EBO sampler uses the same approach but has two Q-tables and applies the Double-Q EBO instead. Pseudo-code for both methods is available in Algorithms 1 and 2.

Code is available at at github.com/pascal314/epistemic-bellman-operators and runs in less than a minute.

Algorithm 1: Pushforward sampling of EBO with i.i.d. ϵ

```
**Input:** Noise-scale $\sigma$
**Output:** A sample from the fixed point

**Initialization:**
$Q \leftarrow \mathbf{0}$
**while** not converged **do**
    $\epsilon \sim \mathcal{N}(\mathbf{0}, \sigma)$
    $Q \leftarrow R + \gamma T^{\pi^*_Q} + \epsilon$
**end while**
**Return** $Q$
```

Algorithm 2: Pushforward sampling of Double-Q EBO with i.i.d. ϵ

```
**Input:** Noise-scale $\sigma$
**Output:** A sample from the fixed point

**Initialization:**
$Q_1 \leftarrow \mathbf{0}$
$Q_2 \leftarrow \mathbf{0}$
**while** not converged **do**
    $\epsilon_{\mathbf{1}} \sim \mathcal{N}(\mathbf{0}, \sigma)$
    $\epsilon_{\mathbf{2}} \sim \mathcal{N}(\mathbf{0}, \sigma)$
    $Q_1 \leftarrow R + \gamma T^{\pi^*_{Q_2}} + \epsilon$
    $Q_2 \leftarrow R + \gamma T^{\pi^*_{Q_1}} + \epsilon$
**end while**
**Return** $Q_1$.
```

Epistemic Clipping PPO (ECPPO)

Code for both Ensemble-ECPPO and Laplace-ECPPO is available at http://github.com/pascal314/epistemic-bellman-operators. We utilize PureJaxRL's implementation of PPO as baseline and backbone of our ECPPO implementations. Pseudo-code for the modified policy update is in Algorithm 4. Acting in the environment and the (clipped) value updates are unchanged from the baseline PPO.

For **Ensemble-ECCPPO**, we replace the value network with an ensemble of 5 value networks with the same architecture. The value update is applied to each ensemble member independently.

For **Laplace-ECPPO**, the value network is unchanged, but now also accompanied by a diagonal approximation of the inverse Fisher information matrix F. Whenever the value model is updated, the inverse Fisher approximation is also updated.

$$F \leftarrow F + \alpha(\frac{1}{\Lambda^2} + n \cdot \frac{g \odot g}{\sigma^2}), \tag{11}$$

where Λ and σ are hyperparameters representing the standard deviation of the prior and likelihood respectively, n is the total number of observations, and $g \odot g$ is the pointwise squared gradients of the value model loss with respect to the previous rollout. Pseudo-code to sample models from a Laplace approximation is provided in Algorithm 3. For a more detailed exposition of the workings of Laplace approximations, we refer to the references cited in the main paper.

Adaptive Clipping For each time step t the uncertainty U_t is computed by taking the standard deviation of the empirically observed advantages, as described in the main text. The clipping parameter of the policy ratio is then modified to $c\phi(U_t)$, where

$$\phi(u) = \frac{1}{2} + \frac{3}{2}\sigma\left(15 \cdot (-u + 0.3)\right),$$

where σ is the sigmoid function.

We hand-picked this function as a simple candidate that maps $[0, \infty) \rightarrow [0.5, 2.0]$, so that ECPPO can either halve or double the clipping range based on the uncertainty. To find reasonable values for scaling and shifting the uncertainty, we conducted an initial experiment on Acrobot-v1 where we empirically evaluated the expected modification to the clipping $\mathbb{E}_U[\phi(U)]$. We then picked the values 15 and 0.3 so that $\mathbb{E}_U[\phi(U)] \approx 1$ on this environment. We did not conduct any other hyperparameter optimization based on algorithm performance to obtain $\phi(u)$. While we used a specifically shaped function, we note that any positive and monotonically decreasing function ϕ is a valid choice.

Algorithm 3: Sampling from a Laplace Approximation

```
Input: Value network parameters $\theta$, diagonal Fisher information matrix $F$ at $\theta$, and number of samples $K$.
Output: Samples $\{\theta_k\}_{k=1}^{K}$ from the Laplace approximation.
for each sample $k = 1$ to $K$ do
    Sample a vector $z_k \sim \mathcal{N}(0, 1)$ with i.i.d. entries, of shape $\theta$.
    Scale $z_k$ pointwise by the diagonal Fisher information
        $z_k \leftarrow z_k / \sqrt{F}$
    Shift $z_k$ by the mean (i.e. the main parameters)
        $\theta_k = z_k + \theta$
end for
Return the set of samples $\{\theta_k\}_{k=1}^{K}$.
```

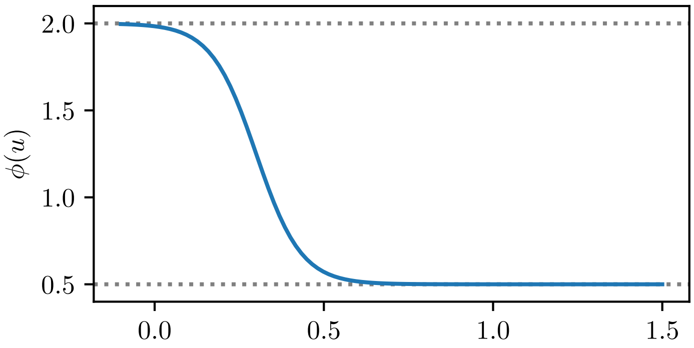

Figure 1: A plot of the function $\phi(u)$ used in our ECPPO experiments.

Hyperparameters We left all further hyperparameters unmodified from the baseline. For completion, we list the hyperparemeters of PureJaxRL's implementation in Table 1.

The learning rate starts at 0.005 and follows a linear schedule down to 0 at the final episode.

The network architectures for both the actor and the value is a fully connected network with hidden sizes 64 and 64, and relu activations. The actor and value networks share no parameters. Epistemic Clipping PPO makes no modification to the actor network, and only replaces the value network with a distributional model.

For Ensemble-ECPPO, this is an ensemble with randomized priors of equivalent architecture, with outputs scaled by a factor of $\beta = 1$, which was not tuned.

For Laplace-ECPPO, this is a Laplace approximation. To find functional hyperparameters, We conducted a small hyperparameter search over $\Lambda = [0.1, 1, 2, 10]$, Fisher learning rate $\alpha = [10^{-2}, 10^{-3}, 10^{-4}, 10^{-5}]$ and $\sigma = [0.1, 1, 2, 10]$ based only on performance of FourRooms. This resulted in prior scale $\lambda = 2.0$, fisher learning rate 10^{-2}, and likelihood $\sigma = 0.1$.

Each baseline and variant of ECPPO ran for 20 seeds on each environment. All experiments were completed on a single GPU, taking around one minute per agent per environment for all 20 seeds in parallel.

Parallel environments	64
Rollout length	128
Epochs	4
Batch size	1024
γ	0.99
Λ	0.95
c	[0.1, 0.2, 0.4]
Entropy loss factor	0.01
Value loss factor	0.5
Max gradient norm	0.5

Table 1: Hyperparameters of the PPO Baseline

Algorithm 4: Epistemic Clipping PPO Policy update

```
**Input:** Initial policy parameters $\theta$, batch of trajectories with $k$ independently estimated advantages $(s_t, a_t, r_t, A_t^{(k)})_{t\geq 1}$. The $k-$th advantage is estimated by the $k-$th ensemble member in Ensemble-ECPPO, or the $k$-th candidate model in Laplace-ECPPO.
**Output:** Optimized policy parameters $\theta$
Normalize advantages:
    $$A_t^{(k)} \leftarrow \frac{A_t^{(k)} - \mu}{s},$$
    with $\mu, s$ estimated from the full batch across all ensemble members combined.
Compute the probability ratio $r_t(\theta) = \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{\text{old}}}(a_t|s_t)}$.
Compute the uncertainties : $U_t = \sqrt{\frac{1}{n}\sum_{k=1}^n (A_t^{(k)})^2 - (\frac{1}{n}\sum_{k=1}^n A_t^{(k)})^2}$   {Empirical standard deviation}
Compute average advantage $A_t = \sum_{k=1}^K A_t^{(k)}$
Compute the objective:
    $$L^{\text{ECPPO}}(\theta) = \mathbb{E}\left[\min\left(r_t(\theta)A_t, \text{clip}\left(r_t(\theta), 1 - c\phi(U_t), 1 + c\phi(U_t)\right) A_t\right)\right]$$
Perform gradient ascent on $L^{\text{ECPPO}}(\theta)$ using Adam
**Return** the optimized policy parameters $\theta$.
```

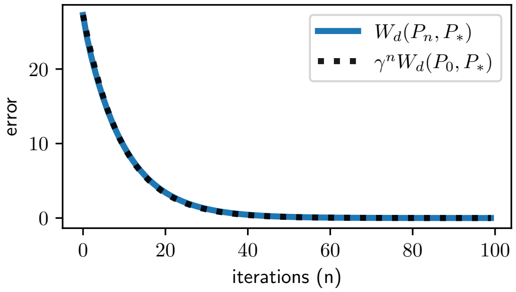

Figure 2: The Wasserstein Distance between distributions over Q-values and the fixed point of the EBO when iteratively applying the EBO (blue). The rate of contraction matches the predicted contraction rate γ (black, dashed)

The regret with respect to the baseline is computed by first identifying the baseline agent that achieves the highest final performance. Then we compute for both ECPPO and the baseline with $c = 0.2$ the regret $R_{\texttt{agent}} = G^* - G_{\texttt{agent}}$ where G^* is the average cumulative reward over the training history (i.e. the area under the curve) for the best baseline, and $G_{\texttt{agent}}$ is the average cumulative reward for `agent`. Finally we report $\frac{G_{\texttt{ECPPO}}}{G_{c=0.2\texttt{baseline}}}$ in a barplot. To verify that ECPPO truly adapts the clipping rate and does not just consistenly increase or decrease it, we color code the environments by which baseline performed best, to highlight that ECPPO can perform well independent of which c was optimal for the baseline PPO algorithm. Code that generated this figure is also available in the supplementary material.

C Additional Figures

Figure 3 shows the learning curves of ECPPO on each environment. It can be seen that in Pong, Acrobot, and FourRooms, Ensemble-ECPPO outperforms all baselines. Furthermore, on MountainCar, Freeway, Breakout and Space Invaders, lower c is optimal in the baselines, and Ensemble-ECPPO matches the best baseline. In Catch, UmbrellaChain, BernoulliBandit, and Gaussianbandit and Cartpole, higher c lets PPO learn faster, and Ensemble-ECPPO again matches the best baseline. In the rest of the environments the relationship between c and the performance of the baselines is not immediately clear, but Ensemble-ECPPO still matches or outperforms the strongest baseline, Except for DiscountingChain, where Ensemble-ECPPO is slightly behind the baseline with the respective c. Laplace-ECPPO also matches or exceeds the baseline in most environments, however, it has a clear disadvantage to the Ensemble-ECPPO and the baselines on Breakout and Asterix. We note that none of the agents solve 15×15 DeepSea reliably, which is unsurprising given that we do not explore actively.

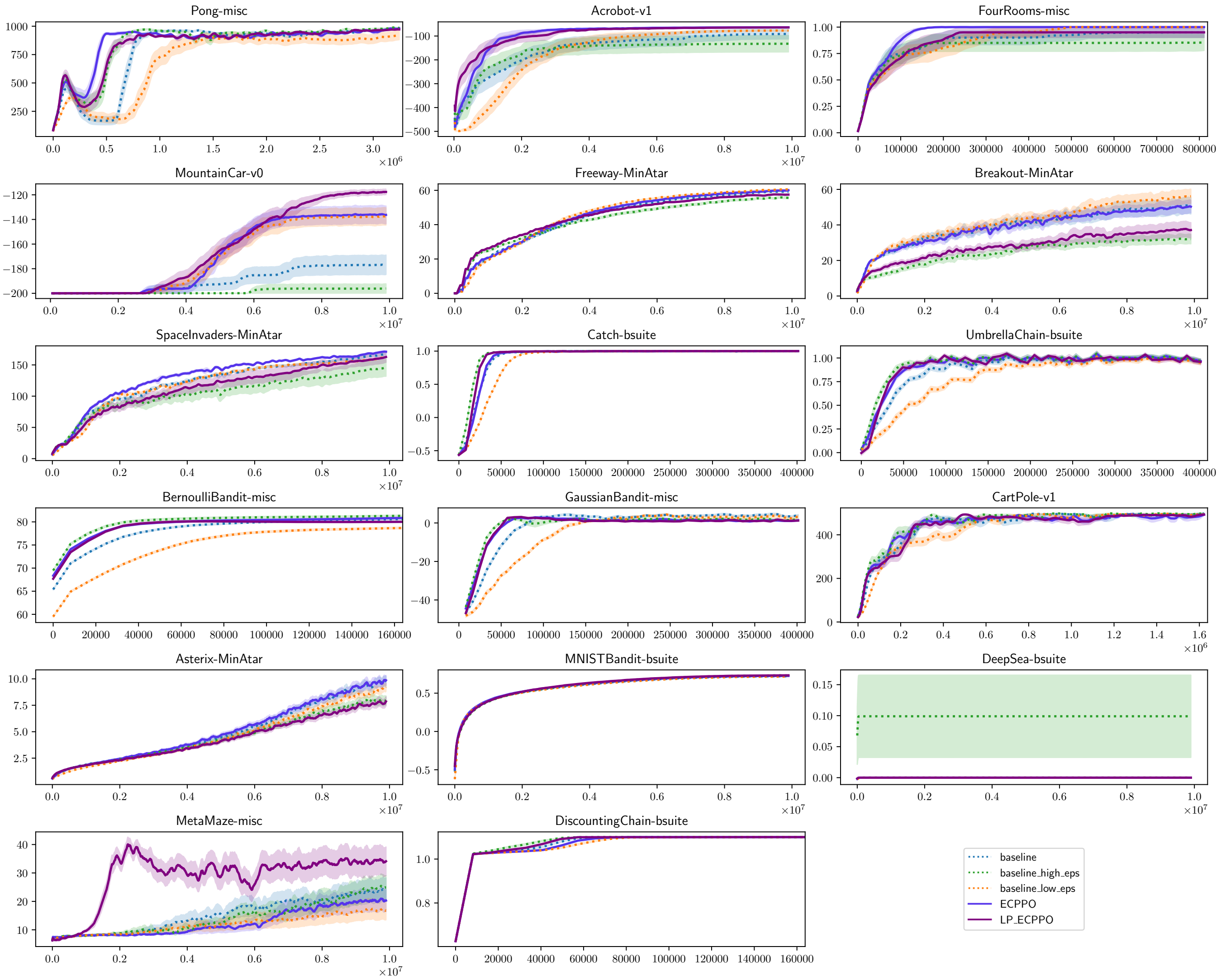

Figure 3: Mean learning curves of ECPPO and baseline PPO algorithms on 17 environments from Gymnax. Shaded areas denote the standard error of the mean, based on 20 seeds per agent per environment.

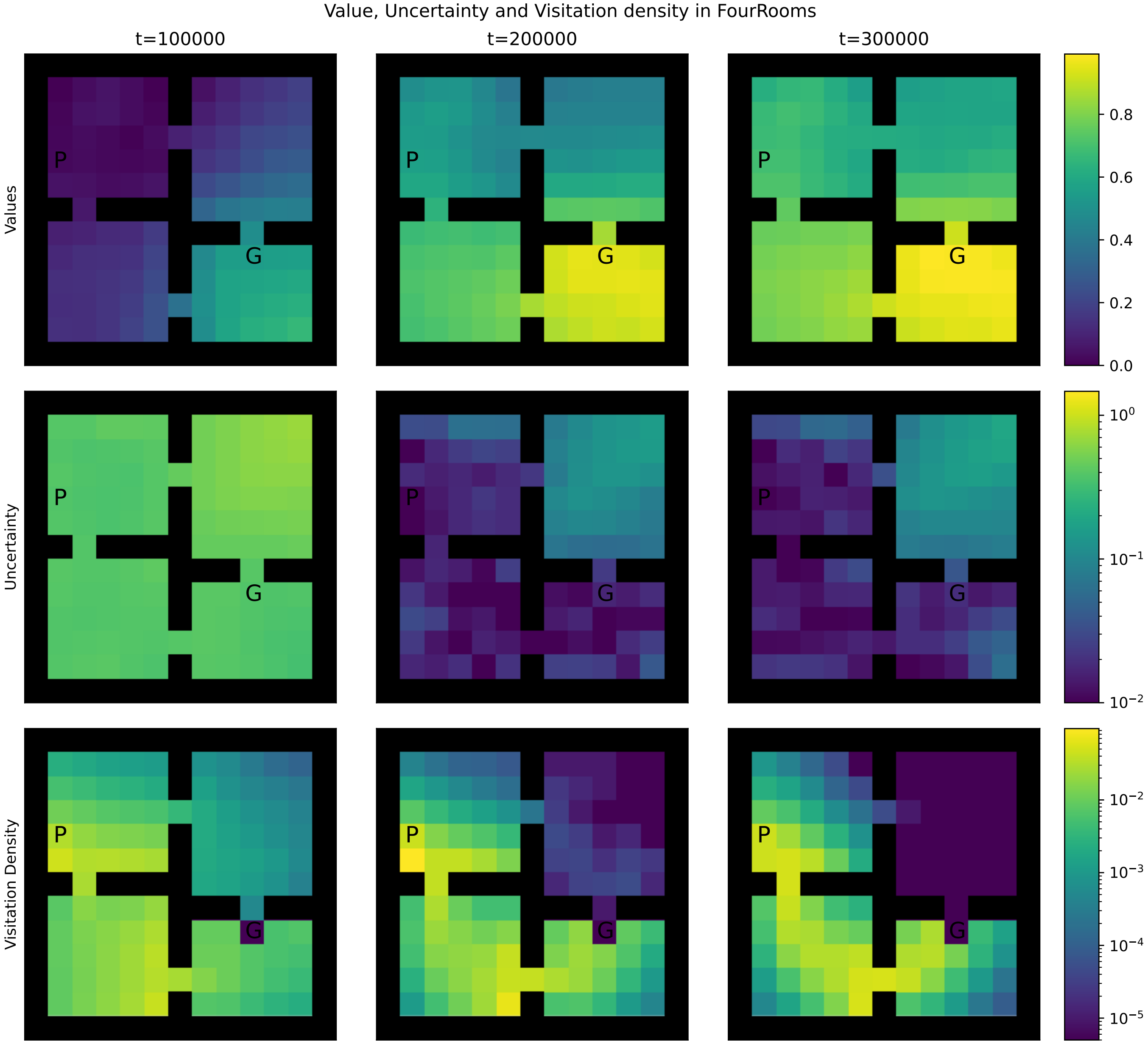

Figure 4: Value, uncertainty and state visitation density on FourRooms of ECPPO at several points during training. The starting position and goal position are denoted by P and G respectively.