A. Implementation Details

Behavior policy We fit the behavior model using a conditional Mixture of Gaussians (Bishop, 1994) with tanh squashing (Haarnoja et al., 2017). We use 5 mixture components. We train the density model with Adam optimizer (Kingma & Ba, 2014) for 10^6 steps and starting from learning rate 10^{-3} and decreasing it by 10 at $8 \cdot 10^5$ and $9 \cdot 10^5$ gradient update steps. Similar to BRAC we train the behavior actor with SAC-style entropy regularization with the same target entropy. We parameterize the model as a 3 layer MLP with relu activations and 256 hidden units.

Actor and critic learning Our implementation is based on Soft Actor Critic (Haarnoja et al., 2019). As in CQL we do not add entropy to the rewards and we modify the critic loss to accommodate the additional regularization term. We use default SAC hyper parameters without additional tuning, in contrast to CQL and BRAC which tune policy learning rate. Following CQL we increased network size for the actor and the critic to 3 layer MLP with 256 hidden units.

Survival bonus The linear term used in CQL can be seen as adding a survival bonus for the environments with early early termination. The derivation is included in the Appendix. Adding a positive constant to the rewards does not have an effect on the optimal policy in infinite horizon MDPs, but in practice Q-targets for terminal states are replaced with 0 that leads to having either a survival bonus or step penalty. For this reason, we add a reward bonus to our implementation as well for fair comparison. We choose the same value $\lambda_{cql} = 5$ as in CQL.

In particular, one can verify that

$$\begin{aligned}
&\nabla_\theta[-\lambda_{cql} Q_\theta(s,a) + (\gamma Q_{\hat{\theta}}(s',a') + r(s,a) - Q_\theta(s,a))^2] = \\
&-\lambda_{cql}\nabla_\theta Q_\theta(s,a) - (\gamma Q_{\hat{\theta}}(s',a') + r(s,a) - Q_\theta(s,a))\nabla_\theta Q_\theta(s,a) = \\
&-(\gamma Q_{\hat{\theta}}(s',a') + [r(s,a) + \lambda_{cql}] - Q_\theta(s,a))\nabla_\theta Q_\theta(s,a) = \\
&\nabla_\theta(\gamma Q_{\hat{\theta}}(s',a') + [r(s,a) + \lambda_{cql}] - Q_\theta(s,a))^2.
\end{aligned}$$

B. Ablations

We also present a full set of results for the ablations in fig. 5.

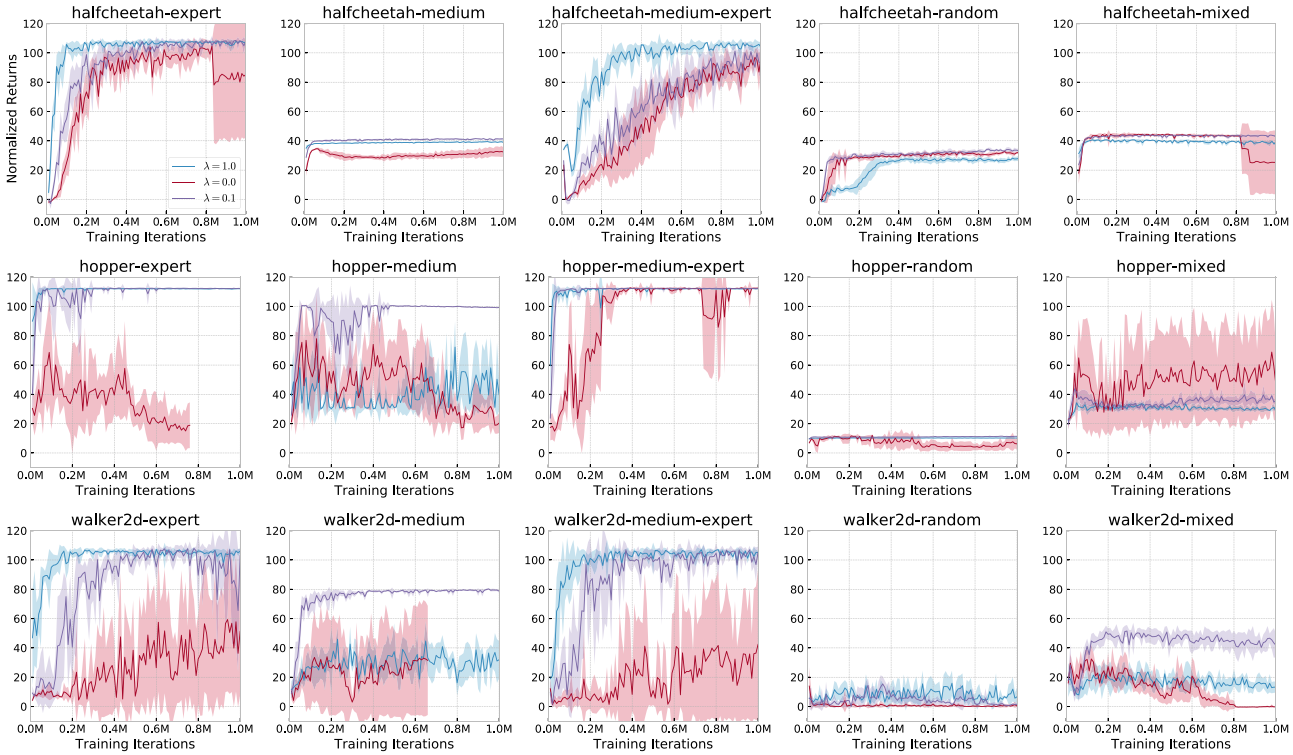

Figure 5. Performance of F-BRC for different values of the gradient penalty coefficient. A larger value, $\lambda = 1$, over-constraints the learned policy to stay close to the behavior policy. This leads to more stable performance on expert datasets, where the behavior policy is near-optimal, but worse performance on medium datasets. Without the regularization ($\lambda = 0.0$) Fisher-BRC collapses on most of these tasks; when the plot is cutoff, it means at least one of the seeds produced NaN values in training.