We thank the reviewers for their constructive feedback. We have reformulated the theoretical statements to position them with previous work, clarified notation and surface-level inconsistencies in theoretical statements, added experiment baselines, and performed an additional comparison with SPIBB (Laroco et al. 2019). The main contribution of our work is a practical offline deep RL algorithm, CQL, which attains strong results and outperforms prior methods by a large margin on a wide range of tasks. We have revised the paper to frame the theoretical results as providing motivation for our approach rather than a primary contribution.

Additional experiments & baselines: Fu et al. (2020) have added results for AWR, BCQ, REM and AlgaeDICE in the D4RL paper, which we now include in Tables 1 & 2. We have added an explicit mention that the baseline numbers are from Fu et al. 2020 [R2, R3, W4] and will add variance measurements [R2]. CQL outperforms AWR, BCQ, REM and AlgaeDICE (R1) in 26/29 tasks, often by a large margin. (R3) We also compared CQL to SPIBB on the Helicopter task from Laroco et al. 2019. With a dataset of size 10k, CQL outperforms SPIBB by attaining 4.11 mean return whereas SPIBB (w/ best Nₐ) attains 3.22 return and soft-SPIBB (w/ best c) attains 3.65 return.

R1/R3, W1: Policy improvement result, what is CQL doing. Based on R1/R3’s requests, we have added a new theorem for policy improvement property of CQL that is more consistent with prior work. Similar to Thm. 1 in Laroco et al. 2019, we show that the CQL updates (Eqn. 2) converge to the optimal policy of a (empirical) penalized RL objective, i.e., πₔ = arg maxₚ Jₚ(π) - αEₚπₚ[Dₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚportion of tasks. We have revised the paper to frame the theoretical results as providing motivation for our approach rather than a primary contribution.

We have clarified these statements (e.g., lines 83-84) to say R1/R3, C2: Offline RL and state distribution shift. We suspect that CQL outperforms robust MDPs empirically due to the approximations necessary to adapt on robust MDPs, CPI, safe PI, high confidence PI, batch RL (Also R1), including 12 references from non-Alphabet authors. We suspect that CQL outperforms robust MDPs empirically due to the approximations necessary to adapt robust MDPs to a practical algorithm. We have also added an empirical comparison to SPIBB.

R3, W3: Relation to prior work. We have added and positioned our ideas with respect to the suggested references for simplicity, we had omitted the count terms from the main text, R1/R3, W1: Policy improvement result, what is CQL doing. Based on R1/R3’s requests, we have added a new theorem for policy improvement property of CQL that is more consistent with prior work. Similar to Thm. 1 in Laroco et al. 2019, we show that the CQL updates (Eqn. 2) converge to the optimal policy of a (empirical) penalized RL objective, i.e., πₔ = arg maxₚ Jₚ(π) - αEₚπₚ[Dₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚportion of tasks. We have revised the paper to frame the theoretical results as providing motivation for our approach rather than a primary contribution.

We have elaborated on this in the paper now. By increasing the difference between Q-values at in-distribution actions (πₐ) and under learned policy (πₜ), CQL constrains the resulting policy to lie in the support of D. This controls the Dₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚₚportion of tasks. We have revised the paper to frame the theoretical results as providing motivation for our approach rather than a primary contribution.