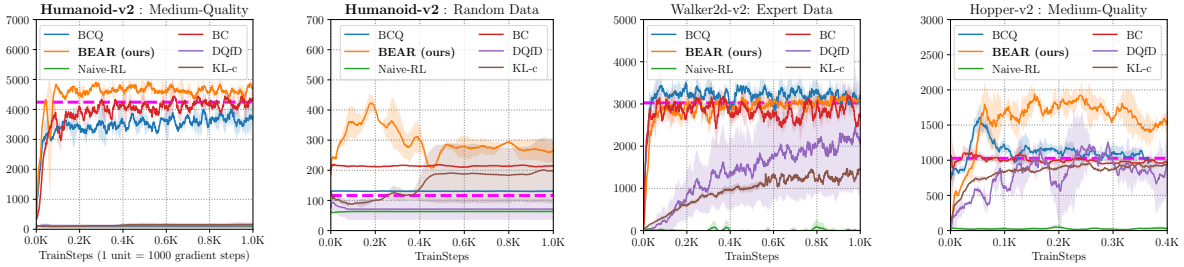
We thank the reviewers for their detailed comments. Our primary contribution is developing theoretical insights and a practical deep RL algorithm (BEAR) for learning from static, off-policy datasets without interaction. The key idea in BEAR is to learn the best policy *within the support of the behaviour/data distribution*.

We have revised the text for clarity, evaluated methods on a more complex task **(Humanoid-v2)** (as requested by **(R1)**), and added two baselines **(R3)**: DQfD (Hester et.al., 2017) (with static data only, as requested by **R2**) and KL-control (KL-c) (c.f., Jacques et.al., 2019). BEAR outperforms all methods on Humanoid-v2, and BEAR outperforms DQfD/KL-c on all other benchmark tasks as well (a subset visualized below). We will release code with the final **(R1)**.



R2: BEAR is not favourable. BEAR is the only algorithm that achieves competitive performance across all dataset compositions. Naive RL fails with optimal data, and BCQ/BC/DQfD/KL-c fail on random data. In practice, most logged datasets are between optimal data and random data ("medium quality"), and BEAR outperforms all methods (BCQ/BC/DQfD/KL-c/Naive RL), often by a *large* margin, in this setting (Fig 3, *orange line*; Figures above).

R2: arguments from lines 127-138: restricting supports vs distributions; comparison to constraint in BCQ We have rewritten the paragraph to clarify the argument. BCQ implicitly constrains the learned policy $\pi(a|s)$ to be close to the behaviour policy β. BEAR, on the other hand, relaxes the constraint to only enforce a support constraint, that is $\pi(a|s)$ has positive density *only where* the density of the behaviour policy is more than a threshold (i.e., $\forall a, \beta(a|s) \leq \varepsilon \implies \pi(a|s) = 0$). $\Pi_\varepsilon \subseteq \Delta^{|S|}$ (where Δ denotes the simplex) is the set of policies, that satisfy this support constraint. Our experiments (Fig 3; Figures above) show that this crucial difference allows BEAR to outperform prior methods especially when the logged data is suboptimal.

R2: Results in Sec 4 lack intuition and connection to BEAR Theorem 4.1 shows a trade-off (lines 172-177) between propagated error and suboptimality bias due to restricting the backups ($\alpha(\Pi)$). In practice, BEAR restricts π_ϕ to lie in the support of β. This insight is formally justified in Theorems 4.1 & 4.2 ($C(\Pi_\varepsilon)$ is bounded). Computing distribution-constrained backup exactly by maximizing over $\pi \in \Pi_\varepsilon$ is intractable in practice. As an approximation, we sample Dirac policies in the support of β (Alg 1, Line 5) and perform empirical maximization to compute the backup. As the maximization is performed over a *narrower* set of Dirac policies ($\{\delta_{a_i}\} \subseteq \Pi_\varepsilon$), the bound in Theorem 4.1 still holds. Empirically, we show this approximation is sufficient to outperform previous methods. This connection is briefly described in Appendix C.2. We now include an explanation of this in Section 5.

R2: How does MMD relate to Π_ε? Directly using MMD would constrain the learned policy to be similar to the behaviour policy in distribution. However, critically, we use a small number of samples to form a sampled MMD estimate. In Appendix C.3, we show that sampled MMD computed from a small number of samples has the effect of measuring support matching, while allowing the relative density on the support to vary. Hence, by penalizing sampled MMD between π and β, we approximately constrain π to Π_ε. The number of samples used for the sampled MMD maps to the threshold ε in Π_ε. Further, for discrete distributions, Gretton et al. (2012)'s example can be adapted to show that sampled MMD with few samples exhibits the desired behavior. We have added this discussion in Section 5.

R1: Principled methods for support restriction We are glad that the reviewer found using MMD for support restriction neat. While this choice is partially justified due to reasons mentioned in Lines 29-32 in this rebuttal, a theoretically robust method is an important next step. We have added a discussion of this in Section 5 as future work and a (theoretical) limitation of our method.

R2: conservative estimate using ensembles; why variance? For the practical algorithm, we used a conservative estimate of Q, as this mitigates overestimated Q-values and leads to improved performance (c.f., TD3). Theoretically, the estimate arises as a high-confidence, lower bound on the true (expected) value via Cantelli's inequality and was used previously with bandits (e.g., CRM, Swaminathan et al. 2015). We have added this intuition and motivation in Sec 5.

R2: λ combination for target values BCQ introduced the idea of using a soft-min $\lambda Q_{min} + (1-\lambda)Q_{max}$. As also reported by BCQ, we find that it performs better than using Q_{min} for the target. We have clarified this in main text.

R2, R3: Baselines: We have added two baselines (DQfD and KL-c), and will add additional baselines in the final if the reviewer has further suggestions. Note, DQfD assumes *optimality* of the static data, which can degrade performance when used with suboptimal data. DQfD, by default, performs online interaction as well.