We would like to thank our reviewers for their thoughtful comments and feedback. We will add the missing citations and fix the typos pointed out by the reviewers.

R1 and R4: *On lack of API description in text.* Our open-sourced repository describes the API in detail, along with examples of agents using it. However, to preserve anonymity, we can not share the link to the repository. Additionally, we will present the API details in the supplementary material of the camera-ready version of our paper.

R1 and R5: *On adding results for online policy selection for the offline policy selection task.* We are concerned that if researchers perform online policy selection and offline policy selection on the same tasks, they may have privileged information and could accidentally bias offline policy selection algorithms towards good hyperparameters. So, we provided distinct tasks for online and offline policy selection in hopes of preventing this. But this choice may be overly cautious, and we are planning to run these experiments since it was suggested by both reviewers.

R1: *On homogeneous data sources.* We agree, offline RL datasets from heterogeneous sources are important, and we would like to include them. The main limitation was a practical one: for challenging tasks training RL agents to solve them, and collecting a large number of human demonstrations are both significant endeavors, and we only focused on one. Our most challenging tasks are locomotion tasks, which are not well suited for human demonstrations. Nevertheless, we believe that the datasets we currently provide are already useful to the community because: 1) they highlight differences between current algorithms, and 2) SOTA algorithms perform quite poorly on the harder tasks. In our tasks, we believe that the main challenge comes from the difficulty of the tasks due to aforementioned properties in our paper. But we believe this is an important direction for research as well.

R1: *On task selection, and defining splits.* Our rationale for choosing this particular task split was to ensure that online and offline tasks both cover easy and hard tasks equally (similar to Atari). We will add this rationale to the paper.

R1: *On architecture used for the BC and other baselines.* We highlight that common control benchmarks in the literature use smaller networks, because they have compact state representations. We use relatively large networks due to the complexity of the DM locomotion tasks. We adopted the same network architecture for consistency, but we agree smaller networks would be sufficient for the control suite tasks. Additionally, we chose to use the batch size of 1024 instead of smaller batch sizes because it runs faster on the hardware we used. Note however that we tried using smaller batch size (256) on a few environments and got identical results (though it took more wallclock time).

R1: *On unconventional GMM parametrization.* Offline RL algorithms should deal with datasets obtained from multiple policies as it would be a common case for real-life applications. Hence, our dm_control and locomotion datasets are generated by multiple agents that often behave quite differently from each other. As a result, we chose to use GMMs since they could better deal with this multimodality in the action space.

R1: *On using BCQ and BRAC public implementations.* We used the public implementations of BRAC from github and its hyperparameters tuned for the dm-control suite tasks. We made sure that our implementation of BCQ reproduces the results in the BCQ paper with the same architecture and hyperparameters.

R1: *On the utility of the "no challenge" dataset.* These datasets are generated from the same tasks as the perturbed RWRL environment, whereas for the control suite we use different tasks. Grouping the “no challenge” dataset together with the combined challenge data for RWRL allows us to examine the effect of the various RWRL challenges on the learning capability of offline RL methods.

R3: *On data from real applications.* We only focus on sim datasets because it is easier to evaluate the resulting policies using publicly available environments. However, working with real-world data requires OPE advances or other ways of scoring the results. We are investigating this possibility and would like to include results from real-world applications in the future. We believe our existing tasks will be useful for developing offline RL methods for real-world applications. Notably, RWRL was designed to include real-world problems such as stuck sensors, delays, perturbations, etc.

R5: *On the lack of details on how rewards are calculated for each tasks.* Our datasets use the original rewards associated with environments that were used to generate the data, and they are described in the documentation of the environments. We will also add these details to the supplementary material of our paper for completeness.

R5: *Goal-conditioned tasks are not included in the benchmark.* Thanks for the suggestion, we are planning to add goal-conditioned tasks in the future, in addition to dmlab and hard-eight datasets. We are open to contributions from the community, and hope that the RL community will contribute new tasks that RL Unplugged is lacking.

R5: *On the effect of data distribution.* We have an analysis of this on Atari and will include it in the supplementary material. We compared different models with respect to the changes in the stochasticity of the transition dynamics, state-action coverage, and reward distributions in the dataset. We found out that behavior regularized models are more robust to those changes.