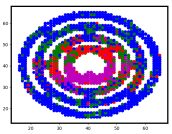
We sincerely thank the reviewers for their helpful comments.

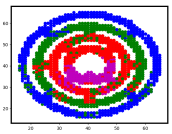
Baselines. The baselines do not solve BiMGame & AntMaze even with optimal trajectories. However, we observe that for BiMGame it is able to get into the 2^{nd} & 3^{rd} ring, and in AntMaze, near the first turn. Fig. D, E shows this as the distribution of visited states (sampled regularly) for value reward (VBRS). We see similar trends for AggreVaTeD. Although they stagnate after making some progress, their cumulative terminal-only reward is 0. (see Line 300-302).

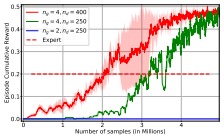
Reviewer 1: • **1a. Order:** We only assume ordering of state groups, which is implicit in many tasks. The trajectories may bifurcate to take different paths to the goal, (as in BiMGame), but our method is able to efficiently learn the subgoals. We empirically show in suppl. that the order assumption is soft, and not strict. Also, it is hardly a limitation to assume the trajectories to start from initial states, as it do not incur an extra cost, and often followed in literature. • **1b. No. of subgoals:** It is a hyper-parameter which we can decide from task info. (Line 257-259) or tuning methods. • **1c. DTW & length:** To assign one of the n_g subgoals to a trajectory state, we perform DTW between subgoals, represented as a sequence of n_g one-hot vectors, and the subgoal pred. p.m.f. of trajectory states (Line 158-160). We do not assume the trajectories to have similar length, for e.g., it ranges in $[53, 850]$ for BiMGame, $[580, 2470]$ for AntMaze. • **1d. One learning process:** Subgoals are only required in the RL step and not to pre-train. One can start directly from RL with subgoals, without pre-training, which will still work, but with many more samples. One can inject the expert trajectories in an off-policy RL method, but it is non-trivial to schedule the sampling of these sub-optimal trajectories. • **1e. Trajectories in other methods:** As the motivation is to show that our method can learn from sub-optimal trajectories, for a fair comparison, we use the same sub-optimal trajectories to generate the sub-goals in our method, to learn the value function in the baselines as well as pre-training the policies in all methods. See "Baselines" above.

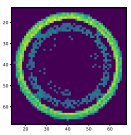
Reviewer 2: • **2a. Perf. of AggreVaTeD & VBRS:** Although subgoal rewards have a similar form as VBRS, the subgoal relabeling step via prediciton+correction helps our method to efficiently learn the subgoals. Fig. B show that without correction using DTW, the subgoals learned are noisy and we find that it cannot learn the task. Also, the strong dependency of AggreVaTeD on the quality of the value func. is discussed in [9, 14]. See "Baselines" & Fig. D, E. • **2b. Different cost func.:** As our expert is a model trained with dense rewards with 'A3C' (Line 228), it is optimal w.r.t. to the dense rewards, but may not be optimal w.r.t. the sparse terminal-only rewards, which we plot in the figures. • **2c. Effect of n_g, necessity of shaping, and $n_g = 1$:** Although Fig. 4a,b show that $n_g >= 2$ works for BiMGame & AntTarget, it does not imply that shaping is not necessary for these tasks. This is because, for $n_g = 1$ (equivalent to A3C in Fig. 3), all states are grouped into a single set, thus no rewards from subgoals, and it fails to learn even with pre-training. Also, Fig. C show that with lesser demos, as pre-training is not as good, $n_g = 2$ fails, but $n_g = 4$ works. • **2d. Fig. 3:** See 2a. • **2e. Fig. 4a:** The drop in the middle for $n_g = 5$ is due to only one of the random RL runs not reaching the terminal for some iter. It is an outlier as the other runs are similar to the other n_g. • **2f. Fig. 4b:** See 2c. • **2g. Mapping.** We meant that in the initial equipartition (Eqn. 2), exactly same states from different demos may have different labels. However, the learned network π_ϕ will always have unique subgoals for each state. We will rephrase. • **2h. Exceeding expert perf.:** Supervised pre-training with expert demos can only achieve the expert perf., albeit with a lot of demos, but cannot surpass the expert [7, 14]. E.g., in BiMGame, we found that models pre-trained with 1500 & 3000 expert demos (of perf. 0.2), perform 0.147 & 0.151 respectively. Our method perf. 0.5 after RL with subgoals. • **2i. Final comment:** Thanks for the great suggestion. • **2j. Non-trivial tasks:** BiMGame and AntTarget are non-trivial tasks as it fails without reward shaping. See 2c. • **2k.**: n_g is inherently related to the task horizon, e.g., $n_g = 4$ works for BiMGame, but not as fast for AntMaze. • **2l. Human-in-loop:** While we do not ask a human for the expert demos (common in literature [8, 9, 14]), the MPC demos can be considered to be similar to human-generated, due to its method of forward simulation in time and selecting the best action.

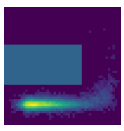
Reviewer 3: • **3a. Use of Neural Networks (NN):** We use NNs to learn the subgoals as the inputs are high dimensional (Line 218, 220) and NN learns the subgoal prediction end-to-end. As suggested, we predict the subgoals using nearest neighbors in high dim. (for BiMGame) and visualize in Fig. A. We also tested our method without DTW and visualize in Fig. B. Both figures are quite noisy compared to our method and using them with RL is not able to learn the task. • **3b. High-dimensional states:** We do learn from high dim. states. The states are 84×84 dim. images in BiMGame and 41 dim. robot state in AntTarget & AntMaze (Lines 218, 220). But, to visualize, we plot the 2D x-y coordinates. • **3c. Trajectory branching:** We do see such trajectories in BiMGame, as there are multiple paths to the goal and trajectories can bifurcate to reach the goal. Results show that our method can learn from such trajectories as well. • **3d. Self-imitation compare:** These works utilize its past experiences to learn faster. However, as it do not use expert demos, to be fair, we do not compare with them and focused on evaluating methods which use expert demos, but we can add this comparison in supplementary.


(A) Using kNN


(B) No DTW


(C) BiMGame


(D) VBRS-BiMGame


(E) VBRS-AntMaze