We sincerely thank our reviewers for the valuable feedback. We note the consensus around the technical novelty of learning compressed representations of the predictive information, the strong empirical performance with comprehensive evaluations, and the clear rationale and presentation. For reproducibility, we plan to release our code by Oct. 1.

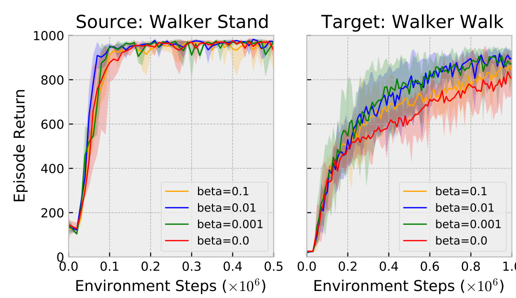
**[R1, R2] Improvements over previous methods and the SotA claim:** Regarding the SotA claim, we will clarify in revision that PI-SAC is better than or at least comparable to any previous SotA for all tasks we evaluated. Additionally, we think the perception that PI-SAC is only slightly better than previous methods is partially a presentation issue. To clarify the differences in performance, the table below is the same PlaNet benchmark comparison table used in both DrQ and CURL. It clearly shows the substantial benefit of PI-SAC. The full table will be included in the revision to augment Fig. 2.

**[R3] Comparison to auxiliary baselines:** We did not include CURL in our submission due to a critical reporting error in the CURL v1 paper (compare the v1 and v3 versions on arxiv). Now that the CURL results have been corrected, we will include them. The table to the right shows that PI-SAC clearly outperforms CURL. Besides MVSP, we also compare to uncompressed PI-SAC since all of the other auxiliary future prediction tasks that we are aware of in the literature do not attempt to explicitly compress the predictive information. In appendix F, we compare to explicit future prediction using generative models and explain that those are also maximizing MI. Finally, as mentioned in Sec. 3, we include future rewards as part of $Y$. We have updated the paper with an ablation removing reward prediction. It slightly degrades PI-SAC performance.

| 100k step scores | PI-SAC | CURL | DrQ |
|---|---|---|---|
| Ball in Cup Catch | **933±16** | 769 ± 43 | 913 ± 53 |
| Cartpole Swingup | **816±72** | 582 ± 146 | 759 ± 92 |
| Finger Spin | **957±45** | 767 ± 56 | 901 ± 104 |
| Reacher Easy | **758±167** | 538 ± 233 | 601 ± 213 |
| Walker Stand | **942±21** | N/A | 832 ± 259 |
| 500k step scores | PI-SAC | CURL | DrQ |
| Cheetah Run | **801±23** | 518 ± 28 | 660 ± 96 |
| Hopper Stand | **821±166** | N/A | 750 ± 140 |
| Walker Walk | **934±53** | 902 ± 43 | 921 ± 45 |

**[R1, R4] Comparing PI-SAC(No Aug) to SAC(Aug) and SLAC:** Sec. 4.2 (from line 142) and Fig. 4 explain why CatGen fails without augmentation. PI-SAC(No Aug) is showing a failure mode; it is not meant to be compared with SAC(Aug) or SLAC. PI-SAC's benefit is demonstrated with the substantial difference between PI-SAC(Aug) and SAC(Aug) in Fig. 3. Also note that SLAC is a completely different system that uses much larger networks and 8 context frames. SLAC's wall clock training time is $\sim$5x slower than PI-SAC. Comparison to SLAC and the other baselines can only be done at a full systems level due to these major differences. It's plausible that SLAC (and Dreamer) would benefit from data augmentation, but PI-SAC would also likely benefit from larger networks and more context frames.

**[R1] Generalization:** In Fig. 7 we mistakenly used different axis scales between figures which obscures the performance difference between source and target tasks. We fixed the axis scales and updated the experiments to use PI-SAC instead of Representation PI-SAC for consistency with the other experiments in the main paper. Results for Walker Stand to Walker Walk can be seen to the right. For dynamics transfer, we varied the testing pole length from 0.4 to 1.6 (trained on 1.0). We find that some compression is always better than none. We will describe these results in the appendix.




**[R3] Choice of DM Control tasks:** The first 6 tasks (out of 9) are the PlaNet benchmark (mentioned in line 108). All of the baselines we compare with use this set. We expanded this set with Walker Stand (for task transfer), Cartpole Balance Sparse (for sparse rewards), and Hopper Stand from the Dreamer benchmark to further explore PI-SAC's generality.

**[R2] Theoretical motivation and generality:** We explore future prediction from an information-theoretic perspective, using CEB [6] to measure and compress the predictive information [4]. As we discuss in Sec. 2, PI-SAC is motivated by the observation that correctly modeling the predictive information requires learning a compressed representation of the past. Due to space limitations, we refer the reader to those works for detailed theoretical background. In Sec. 5 (line 193), we list previous successes of future prediction for representation learning and auxiliary tasks on various types of RL problems, which is evidence that PI-SAC should apply more broadly.

**[R1, R2] Representation dependence on policy and choice of X and Y:** The CEB model captures only the environment dynamics $s, a \rightarrow s'$ (which is independent of the policy) by conditioning the encoder $e(z|x)$ on the future actions (actions are part of $X$). Part of this is explained around lines 201-204, but we will add clarifications. Following CURL and DrQ, we use 3 frames for $X$. We make $Y$ symmetric to $X$; it contains the next 3 frames and their rewards.

**[R1, R2, R4] Citations and other clarity questions:** Thanks for suggesting the curiosity papers – we will include them in Sec. 5. We will make the CEB and PI-SAC descriptions more self-contained, and improve Sec. 3. We will add descriptions for double critics (they are part of SAC [11]) and improve the notation. We updated the generalization section to use PI-SAC rather than Representation PI-SAC (see the Walker task transfer figure above).