We would like to thank all the reviewers for taking time to read our paper and providing valuable suggestions. We're happy to see that reviewers like our paper and we provide new experiments with explanations to address their concerns.

R3: No comparison to RNN architectures which create natural bottlenecks. Thank you for this suggestion. We now replace the fully-connected history-stack encoder network in our BC-MO baseline with an RNN sequence encoder of similar size and output dimension. The table below shows episode rewards for the original fully-connected BC-MO vs. this new RNN variant. RNN scores are mostly on par with BC-MO, and much worse in Ant and Walker2D envs.

	PO-Ant	PO-Hopper	PO-Humanoid	PO-Reacher	PO-Walker2d	PO-HalfCheetah
BC-MO (Feed-forward)	1750 ± 146	293 ± 83	565 ± 80	-64 ± 4	592 ± 124	820 ± 60
BC-MO (RNN)	-311 ± 150	315 ± 32	367 ± 64	-75 ± 5	190 ± 14	830 ± 398

R3: Other techniques contending with partial observability, missing citations. Thank you for this suggestion; we'll set up this broader context in introduction, and include a paragraph in related work.

R1: Why TRPO as expert demonstrations? Expert data too "clean". As R1 notes, using an RL agent as the expert is standard in imitation learning, e.g. DAgger, GAIL, and CCIL, which is the most closely related prior work to ours. We have now generated new noisy demonstrations by executing the trained RL expert in ϵ-greedy exploration mode ($\epsilon = 10\%$), sampling exploration actions from $U[-1, 1]$. Although the expert as well as all imitators perform worse now, our method still performs slightly better than BC-MO. For example, in Reacher, reward is -68 ± 3 for BC-MO vs. -61 ± 10 for ours; in HalfCheetah, reward is 97 ± 29 for BC-MO, vs. 154 ± 82 for ours. We will include results from all 6 environments in camera-ready.

R1&R4: Results in high dimensional and naturally partially observed environments. We have now performed two new experiments on Atari Enduro and UpNDown. As in CCIL, we use a β-VAE to encode images. We use 10k transitions for Enduro and 200k transitions for UpNDown. Our method outperforms BC-MO and CCIL.
In Enduro, the rewards are **Ours**: 27 ± 1, BC-MO: 24 ± 4, CCIL: 13 ± 2 . Expert: 52 ± 1.
In UpNDown, the rewards are **Ours**: 54 ± 3, BC-MO: 50 ± 2, CCIL: 26 ± 6. Expert: 64 ± 4.
Note: the original CCIL paper experiments don't use observation histories, and CCIL struggles with these higher dimensional inputs.

R4: CCIL with more interactions? Compared to our approach, CCIL requires *additional* environmental interactions aside from the demonstration data. In the paper, we use a comparable number of interactions to the original CCIL paper. We have now increased the number of interactions from 100 to 1000 for Hopper, improving the reward from 144 to 224, but still much poorer than ours (1086). With even more interactions and a well-disentangled representation, CCIL may be able to eventually outperform ours, but as R4 points out, that would not undermine our purely offline approach.

R2: Why easier to infer past actions than the next action? Always true? Indeed, this is not always true, and we did find environments where behavior cloning from observation histories did not manifest the copycat problem, e.g., Atari Pong. More broadly, inferring past actions is an example of a "shortcut", as R4 points out. As Geirhos et al, "Shortcut Learning in Deep Neural Networks" mentions, it remains an open problem why neural networks find some "shortcut" solutions easier to learn, compared to the "correct" solutions, but this is an interesting direction for future research.

R2: Information bottleneck ad-hoc? Theoretical justifications? The information bottleneck (IB) demonstrably contributes to our method's performance (see paper Tab 2). Conceptually, our approach is built around identifying observation histories as likely to contain nuisance information. The IB provides a natural way to penalize information transmitted from this history. IB has been used in other works in similar ways, e.g. Pacelli 2020, "... Task-Driven Control ...", and Rakelly 2018, "... probabilistic context variables". For theoretical justifications for IB, see Alemi et al "Deep variational IB". We will motivate IB better in camera-ready and add these related works.

R1: Error bars on action predictability. The updated results with error bars are shown in the table below. We will add error bars to Tab 3, 4, and 5 in camera-ready.

R2: "Copycat" problem or just averaging out noise? Compare BC-SO? Thank you for this perceptive comment. BC-MO observes history, so it obtains full information and BC-SO is not comparable with it. We find it more reasonable to compare with BC-SO (Full state) as it has the same information as BC-MO and, as suggested by R2, both of them would suffer from the "averaging out". In the table below, we show that the next action is more predictable in BC-MO than in the history-independent BC policy, suggesting that "copycat" problem exists in these environments. We will clarify in camera-ready.

	Ant$\times 10^{-2}$	Hopper$\times 10^{-3}$	Humanoid$\times 10^{-1}$	Reacher$\times 10^{-5}$	Walker2d$\times 10^{-2}$	HalfCheetah$\times 10^{-2}$
expert	6.91 ± 0.21	8.60 ± 1.09	6.93 ± 0.32	1.46 ± 0.37	2.47 ± 0.07	9.81 ± 0.33
BC-SO (Full State)	3.56 ± 0.07	2.55 ± 0.72	6.32 ± 0.75	0.33 ± 0.05	0.96 ± 0.00	3.32 ± 0.11
BC-MO	0.66 ± 0.04	1.07 ± 0.16	0.18 ± 0.01	0.32 ± 0.05	0.46 ± 0.02	2.97 ± 0.15

R3: Figure 3 needs a line of best fit and R^2 value and explain the outliers. We fit the curve with an inverse proportional function, yielding $R^2 = 0.74$. To clarify, we do not claim that action predictability is the sole determinant of reward, just one factor. Action predictability is a symptom of the copycat problem, but it is likely also influenced by the nature of the specific task and demonstrations. As such, while the overall trends are clear, it is difficult to explain outliers.