Reviewer 1 Thank you for your detailed and in-depth review! *Comparison with other ways of mixing:* In preliminary experiments we tried a schedule in which noise-injection was only introduced after an initial burn-in period. However, the results were much worse, suggesting that it is important for learning good representations that the network is trained on the noisy term from the very beginning. On Coinrun we show that both terms in eq. 7 together substantially outperform using either one of them, supporting the choice we made for SNI. We agree that many other (e.g. continuous) scheduling approaches might also be feasible and we hope to encourage further research in this area. However, we believe we sufficiently show the advantage of our method compared to prior published work by comparing to the existing state of the art for both domains.

Loose notion of selectivity: We would like to note that in the mode *with* noise injection, it is done so selectively (i.e. not for V and π_θ^r), thus motivating our choice of name. *Motivating eq 7 from section 3*: Issue i) Noise injection can make the model more robust by inducing mistakes that are then improved by gradient updates. However, those mistakes are not useful when acting where they can induce bad actions, so we use the no-noise policy for rollouts $\bar{\pi}_\theta^r$. Note the faster learning of IBAC-SNI vs. IBAC in Fig. 6 (appendix). ii) Using noise only for updates can lead to a high IS variance, therefore we use the mixture, which empirically improves results (IBAC-SNI with $\lambda = 1$ (no mixture) vs $\lambda = 0.5$). iii) The additional noise through the critic is eliminated in the policy update by using $\bar{V}_\theta$. We will emphasise these links stronger in the revised paper. *Deep analysis:* We agree and will include plots for variance terms (please see Fig. 1 below for more results on Multiroom, in addition to Fig 3-right already in the paper) and link to videos of trajectories.

Minor comments: Those are very helpful, thank you! $\mathcal{Z}$: you're right. Semi-gradient: Yes, we agree this is a better description than "assumed non-changing". L_{AC}^V and $V_\theta(z)$: You're right, we will properly introduce them. L_{AC}^V is eq. 6 and $V(z)$ is the value function taking the latent z as input which depends on s through the encoder. Fig 3: Yes, in the updated figure below. No qualitative change in results. 'Outperforms equal mixing': Yes, it refers to Dropout only.

Reviewer 2 Thank you for the insightful feedback! We completely agree and will be more explicit about the specific setting we are investigating in the paper, including re-formulating the statements you mentioned. We agree that seeing RL as just minimizing some loss is overly simplistic. We chose this perspective to highlight differences to supervised learning, but will be more nuanced in the updated version of the paper. *Minor points:* Thank you! We will make these changes. Line 233: Yes, successful agents achieve a return of close to 1. We hope the additional plots in Figure 1 show the results better. They also now use 30 instead of 5 seeds on Multiroom and at least 3 seeds on Coinrun.

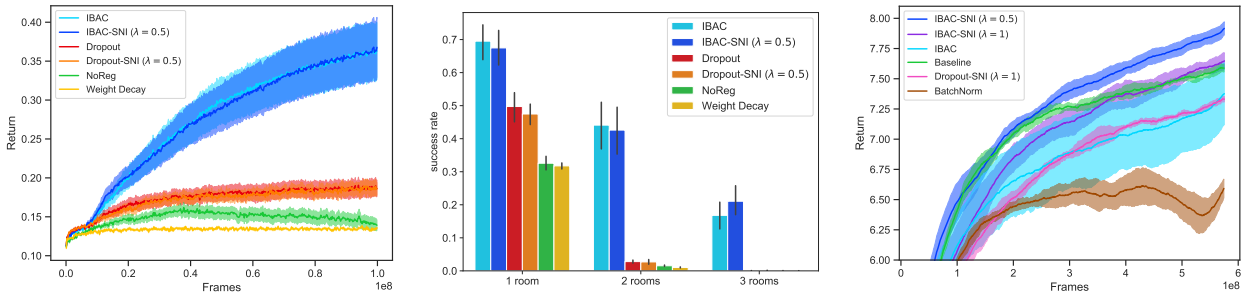

Figure 1: More seeds and additional plot for Multiroom: In addition to the average return (left), it shows the probability of successfully finding the goal in a layout with a certain number of rooms after training (middle). Note that models are still trained on a mixture of rooms and this separation is only done for testing. The results are averaged over 100 test-layouts per seed and 30 different random seeds per algorithm. The error bars are across seeds. The middle figure shows how much IBAC outperforms alternative regularization techniques especially for more difficult layouts. This shows it allows learning better representations faster and with less available data. Note the *same* model is tested across different room numbers. Especially for more complex layouts (3 rooms) the added stability provided by SNI starts to improve performance. Nevertheless, the main difficulty in multiroom is in learning *general representations*, highlighting the utility of IBAC to do so. SNI becomes necessary on more complex environments like Coinrun (right) where SNI is critical for SOTA performance (note e.g. that multiroom has no 'catastrophic' actions that end the episode). Coinrun results are averaged over 3 seeds (like in [10]) as they are expensive. We will have at least 5 seeds in a future version.

Reviewer 3 Thank you for your positive and constructive review! Line 71 and Equation 5 have been corrected as you suggested. Line 95: Yes, Dropout was applied only on the last layer in all experiments. We found multiple architectures in the literature but this seems to be the most often used. We will explicitly state this. Equation 12: λ, λ_H and λ_V are distinct and we agree that a more distinct notation would be better and will update the paper. The values are given in the appendix but we forgot to mention that λ_H and λ_V are shared across all experiments and algorithms using the widely used values $\lambda_H = 0.01$ and $\lambda_V = 0.5$. Yes, L_{AC}^V is indeed equation 6. In practice we use PPO, i.e. L_{PPO}^V, equation (2). We will clarify the presentation here.