We thank the reviewers for their comments. We will address all style suggestions and minor points.

Explaining limitations of our approach: The main comment in all 3 reviews is the suggestion to include more discussion or experiments with small problems to illustrate the limitations of the approach, which we agree is a good idea. First note that the main assumptions of SNAP are the independence assumption in rollouts and the fact that the rollout plans do not depend on observations beyond the first step. This last restriction can be lifted at the cost of being exponential in depth, as in other algorithms, but as experiments show speedup is crucial in large problems.

Deterministic transitions are not necessarily bad for factored representations because a belief focused on one state is both deterministic and factored and this can be preserved by the transition function. Both the T-maze and the rocksample domains that were proposed in the reviews are actually suitable for SNAP. The reason is that one step of observation is sufficient and the reward does not depend in a sensitive manner on correlation among variables. The [Pajarinen] paper shows that factoring works well for rocksample. We encoded the T-maze domain in RDDL and show the results in the table. We use 3 seconds per step for each algorithm, and the T-maze length is shown in the first row. YES means that the algorithm finds optimal actions for all steps, NO means the algorithm does not, and X means the result is not available as DESPOT only runs with pomdpx and the RDDL to pomdpx translator failed for this problem size. The result shows that SNAP can solve the T-maze problems.

		5	10	12	15	30	50
SNAP	YES	YES	YES	YES	YES	YES	YES
DESPOT	YES	YES	YES	X	X	X	X
POMCP	YES	NO	NO	NO	NO	NO	NO

However, we can illustrate the tradeoff with two other simple domains. The first has 2 states variables x_1, x_2, 3 action variables a_1, a_2, a_3 and one observation variable o_1. The initial belief state is uniform over all 4 assignments which when factored is $b_0 = (0.5, 0.5)$, i.e., $p(x_1 = 1) = 0.5$ and $p(x_2 = 1) = 0.5$. The reward is if $(x_1 == x_2)$ then 1 else -1. The actions a_1, a_2 are deterministic where a_1 deterministically flips the value of x_1, that is: $x_1' =$ if $(a_1 \wedge x_1)$ then 0 elseif $(a_1 \wedge \bar{x}_1)$ then 1 else x_1. Similarly, a_2 deterministically flips the value of x_2. The action a_3 gives a noisy observation testing if $x_1 == x_2$ as follows: $p(o = 1) =$ if $(a_3 \wedge x_1' \wedge x_2') \vee (a_3 \wedge \bar{x}_1' \wedge \bar{x}_2')$ then 0.9 elseif a_3 then 0.1 else 0. In this case, starting with $b_0 = (0.5, 0.5)$ it is obvious that the belief is not changed with a_1, a_2 and calculating for a_3 we see that $p(x_1' = 1 | o = 1) = \frac{0.5 \cdot 0.9 + 0.5 \cdot 0.1}{(0.5 \cdot 0.9 + 0.5 \cdot 0.1) + (0.5 \cdot 0.9 + 0.5 \cdot 0.1)} = 0.5$ so the belief does not change. In other words we always have the same belief and same expected reward (which is zero). Therefore, for this problem factoring implies that the search is blind. On the other hand, a particle based representation of the belief state will be able to concentrate on the correct two particles (00,11 or 01,10) using the observations.

The second problem has the same state and action variables, same reward, and a_1, a_2 have the same dynamics. We have two sensing actions a_3 and a_4 and two observation variables. Action a_3 gives a noisy observation of the value of x_1 as follows: $p(o_1 = 1) =$ if $(a_3 \wedge x_1')$ then 0.9 elseif $(a_3 \wedge \bar{x}_1')$ then 0.1 else 0. Action a_4 does the same w.r.t. x_2. In this case the observation from a_3 does change the belief, for example: $p(x_1' = 1 | o_1 = 1) = \frac{0.5 \cdot 0.9}{0.5 \cdot 0.9 + 0.5 \cdot 0.1} = 0.9$. That is, if we observe $o_1 = 1$ then the belief is $(0.9, 0.5)$. But the expected reward is still: $0.9 \cdot 0.5 + 0.1 \cdot 0.5 - 0.9 \cdot 0.5 - 0.1 \cdot 0.5 = 0$ so the new belief state is not distinguishable from the original one, *unless one uses additional sensing action a_4 to identify the value of x_2*. In other words for this problem we must develop a search tree because one level of observations does not suffice. If we were to develop such a tree we can reach belief states like $(0.9, 0.9)$ that identifies the correct action and we can succeed despite factoring, but SNAP will fail because the search is limited to one level of observations. Here too a particle based representation will succeed because it retains the correlation between x_1, x_2.

Rev1 - Heuristic domain knowledge: we agree that the performance of MCTS will improve with domain specific knowledge. However, our focus was on domain independent performance of the planners. In addition, since different algorithms might use domain knowledge in a different manner the comparison of algorithms would be less clear.

Rev1 - Prior work with factored belief: Thanks! We will add a discussion of this and other work to the paper.

Rev2 - Evaluating contributions from SOGBOFA: We agree that it is important to understand the contribution of different components. Contributions of components that are part of SOGBOFA were reported in the cited papers, albeit in the context of MDPs. The experimental evaluation in this paper is centered on evaluating the the new ideas in this paper, showing the importance of sampling for large observation spaces, and performance sensitivity w.r.t. run time per step and number of samples.

Rev2 and Rev3: line 96: Yes $p(x = 1)$ should be $p(x = T)$. We will plan to improve the description. In the current notation, for a generic variable x assume $p(x = T)$ is given by the RDDL expression if $(cond1)$ then p_1 elseif $(cond2)$ then p_2 else p_3 where the conditions form a set of mutually exclusive and exhaustive set of conjunctions. Then the probability of x given that $cond_i$ holds is p_i. In general, assuming discrete variables, the CPT of the variable x can be rewritten in this form and this is facilitated by the RDDL representation.

Rev3 - plan vs. policy: We will plan to improve the description. By a plan we meant a pre-determined sequence of actions not conditioned on states or observations. The same notion is also known as an open loop policy and as a straight line plan. A policy will condition future action choices on the future states (in MDP) or belief states (in POMDP).