We thank the reviewers R1-R4 for their insightful comments. We are happy to note the reviewers agree on the novelty, correctness, and relevance of our work. Comments raised are addressed below and will be clarified in a final version.

Motivation for the proposed approach, centralized baselines (R2, R4). Decentralized active perception problems may arise in applications such as a team of robots executing an information gathering mission in a remote environment, e.g., search and rescue, or situational awareness applications. The robots plan and coordinate their actions centrally before task execution, but plan execution is decentralized as communication or otherwise sharing observations is not possible during task execution. This problem is formulated as Dec-ρPOMDP with a reward that is a convex function of the joint state estimate in Lauri et al. (2019), which prevents applying standard Dec-POMDP algorithms that assume a reward that depends on the hidden state. The motivation of our work is to enable application of standard Dec-POMDP algorithms to multi-agent active perception by reducing the problem to a Dec-POMDP with a standard reward.

The Dec-ρPOMDP we target is computationally more challenging (NEXP-complete) than centralized POMDP, POMDP-IR, or ρPOMDP (PSPACE-complete). Centralized variants also make fundamentally different assumptions: all-to-all communication without delay during task execution would be required to solve the problem we target as a centralized problem, and the resulting optimal value would be higher (Oliehoek & Amato, 2016) as such approaches exploit knowledge centrally. All these differences mean that a comparison to centralized methods is neither fair nor necessary.

Comments from R1. The table below shows the average policy value $\pm$ its standard error, over 100 runs. Additionally, we observed the best value over all runs for APAS to be similar to NPGI average, showing the potential of our method. The updated results will be included in the final paper. Generally, scaling Dec-POMDP solutions to large horizons is difficult, and we believe the optimization is struggling in these cases. Due to our paper, however, any improvements to Dec-POMDP optimization can transfer to the active perception setting.

	$h=2$	$h=3$	$h=4$	$h=5$	$h=6$	$h=7$	$h=8$
MAV, APAS (Ours)	-2.006 $\pm$ 0.005	-1.936 $\pm$ 0.004	-1.879 $\pm$ 0.004	-1.842 $\pm$ 0.005	-1.814 $\pm$ 0.004	-1.789 $\pm$ 0.004	-1.820 $\pm$ 0.005
MAV, APAS (no adapt.)	-2.124 $\pm$ 0.004	-2.000 $\pm$ 0.006	-1.936 $\pm$ 0.007	-1.913 $\pm$ 0.006	-1.899 $\pm$ 0.007	-1.882 $\pm$ 0.008	-1.901 $\pm$ 0.008
Rovers, APAS (Ours)	-3.496 $\pm$ 0.005	-3.416 $\pm$ 0.007	-3.337 $\pm$ 0.011	-3.317 $\pm$ 0.010	-3.322 $\pm$ 0.013	-3.358 $\pm$ 0.012	—
Rovers, APAS (no adapt.)	-3.796 $\pm$ 0.020	-3.718 $\pm$ 0.023	-3.613 $\pm$ 0.019	-3.602 $\pm$ 0.023	-3.644 $\pm$ 0.025	-3.707 $\pm$ 0.028	—

Time per improvement step (MAV, $h=5$) was 1.1 seconds (APAS) versus 30 seconds (NPGI). Problem creation is fast in domains we study, detailed results will be added. Bounding ρ from below and above is required for Thm. 1 to hold.

Comments from R2. Motivation of the work is addressed above. We disagree that our approach is essentially heuristic: if required, a bounded error is obtained applying an exact or approximate solver, see also comments for R3 on approximating final reward. We apply the *finite horizon* variant of Pajarinen & Peltonen, (2011, Sect. 3.1.1), as it is similar to the current state-of-the-art algorithm (Lauri et al., 2019) for Dec-ρPOMDPs. We agree that scalability w.r.t. number of agents and size of the problem is interesting, but to some extent orthogonal to the goal pursued in this paper: being able to treat active perception problems as standard Dec-POMDPs. We note, however, that our reduction of Dec-ρPOMDP to the standard Dec-POMDP lays a solid theoretical foundation to investigate scalability in future work.

Our current results demonstrate our approach is reasonable and increases scalability in the planning horizon. We argue above why a centralized baseline does not provide a fair comparison. While interesting further results *could* arise, e.g., from applying different Dec-POMDP solvers or RL methods, at the same time it is not clear what further information we could expect from them. Further, given the substantial amount of effort (e.g., the JAIR 2016 algorithm by Dibangoye et al. is extremely complicated, no code is publicly available, and we estimate that implementing and debugging would take months) and energy that would be required (deep learning approaches need large amounts of data and time), we do not think that this is a reasonable request.

If R2 insists more baselines are required, we would hope to see in an updated review a motivation of what these additional experiments would intend to demonstrate. Otherwise it is difficult to improve our paper.

Comments from R3. We will revise notation to show the correspondence of the problems and value functions, and review the notation to see if we can further clarify. Further suggestions are also much appreciated. Convex final rewards (e.g., negative entropy) are approximated more accurately by a greater number of α-vectors: a detailed error bound is found in Sect. 4.1 of Araya-López et al. (2011). The bound also applies to our paper. We initialize sets of α-vectors by first drawing corresponding joint state estimates randomly (Lines 297-300 in paper). Performance depends on selection of the α-vectors, as seen by APAS performing better than the "no adaptation" version with fixed α-vectors.

Comments from R4. Please see above for clarification on the motivation of the work.