We thank the reviewers for all of these valuable comments. We provide point-by-point responses below.

Re: generalize to other applications. The target coverage problem is a fundamental and challenging problem in most DSNs, covering various practical applications, such as camera networks for sports game videos capturing and directional radar networks for aircraft tracking. We formulated it as a Multi-agent Markov Dynamic Process and proposed a coordinator-executor mechanism with a bunch of practical methods. We are grateful for your reminder of the generalization potential of our approach. Indeed, it is promising to apply the coordinator-executor hierarchy to other environments with multi-targets and multi-agents. Upon your suggestions, we tried to apply our mechanism in the 3v3 Cooperative Navigation problem (Lowe et al. '17) and achieved a competitive mean reward (-4.8) against MADDPG (-5.3) and COMMA (-7.7). It is an interesting future direction to apply our method on other multi-target multi-agent coordination problems.

[R1] Q1: training method. We found that simultaneously training the coordinator and executor works poorly, as they destabilize each other. Specifically, the stochastic target selection will make the executor inefficient to learn. Meanwhile, the poor executor will lead to a noisy global reward, destabilizing the learning of the coordinator. Instead, the two-step training strategy could avoid these problems. The coverage rate of the end-to-end method is 36.57 ± 6.78, which is much worse than the two-step training.

Q2: reward design. The reward function design is based on the domain knowledge of the DSNs, counting the coverage rate and the energy cost. We will further discuss the factors of each component in the next revision.

Q3: scalability. We have compared the scalability of our method with ILP in the case of 3-7 targets and 2-6 sensors (See L288-297 and Fig.4(b)(c)). Notably, our model is only trained in a specific setting (4 sensors, 5 targets). If the evolutionary population curriculum approach (Long et al. '20) is employed while training, the scalability could be improved further.

Q4: the critic based on global feature. This claim is supported by the poor performance of the case (2) in ablation study, described around L280. Main reason is that the global feature C_t just aggregates the local features θ_t^i by sum. While, the critic based on AMC can assign credit more accurately, which leads to a better policy.

Q5: training episodes for the baselines. Sorry for the typo. In fact, all methods are equally trained with 50k episodes.

Q6: clarity: 1) In that paragraph, we want to argue that the order of relation observations wouldn't matter, because the relations are pairwise and unordered, not like text data requiring sequential processing. So, the attention mechanism is better than RNN for the coordinator to encode an order-invariant representation. Then, IDs are just for distinguishing relations from each other. 2) The role of the **filter** is to remove the redundant target-agent relations from the observation $o_{i,t}$ of the executor i according to $g_{i,t}$. For example, if $g_{i,t}$ is $[1, 0, 1]$ and $o_{i,t}$ is $[o_{i,t}^1, o_{i,t}^2, o_{i,t}^3]$, then $f(o_{i,t}, g_{i,t}) = [o_{i,t}^1, o_{i,t}^3]$. In this way, the executor can ignore irrelevant targets and focus on the assigned targets. 3) The typos and formatting error would be corrected in the next revision. Thanks for pointing them out.

[R2] Q1: It is not clear that the application area calls for a multi-agent model here. For a sensor network, it is necessary to consider the communication cost among sensors. Thus, a fully-centralized controller is not a good solution to this problem. In our mechanism, the coordinator is only allowed to communicate with executors **periodically (every 10 steps)**, instead of every step. And each distributed executor only needs to focus on its sub-task independently without any additional communication. Besides, the coordinator could pay more attention to long-term schedule.

Q2: only compare with other MARL approaches? In fact, we already compared our method with a non-MARL baseline, namely ILP (Integer Linear Programming). Please refer to Appendix.2 for the implementation details and further analysis. Such a fully-centralized optimization method requires a strictly problem-specific formulation and some fine-crafted constraints, but performs worse than our MARL method. This also shows the necessity of a multi-agent model for this application (mentioned in **R2Q1**). Upon your suggestion, we also ran the trivial distributed heuristics. The performance [coverage rate: $60.65 \pm 7.91\%$, average gain:1.63] is worse than ours.

[R4] Q1: standard benchmarks. In this paper, we aim at finding a solution to address the target coverage problem in DSNs and new challenges encountered. We evaluate methods on the customized environments as there is no benchmark that could satisfy our problem setting. If accepted, we will release our environments to enrich the public benchmarks and attract researchers to pay more attention to the new challenge, i.e. multi-agents multi-targets assignment.

Q2: the technical novelty. 1) We first propose a Hierarchical Multi-agent Coordination Mechanism for solving the target coverage problem in DSNs. There is no prior work tackling this particular challenge via RL (as mentioned by R2). 2) Then, we introduced a bunch of practical and effective methods for the mechanism. Particularly, "The framing of the higher-level hierarchy estimating the marginal contribution of each agent is original and interesting." (as recognized by R1) 3) It is also promising to generalize our approach to other domains (as mentioned by R1).

Q3: the average gain This is an auxiliary metric to evaluate the efficiency of the energy consumption. It is only useful when a competitive coverage rate has been achieved. COMA only learns to take no-operation most of the time, so it saves the energy but also get the worst performance in the coverage rate. ILP prefers to rotate the sensors only when the targets are being out of the current direction partition as analyzed in Appendix.2, so its average gain is the highest.

Q4: release the codes/environments? If accepted, we will release the environments, codes, and trained models.