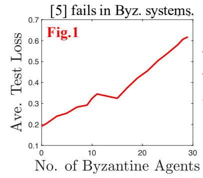
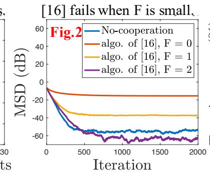
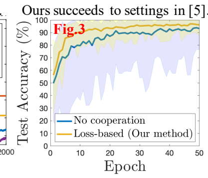
We thank all the reviewers for their valuable comments and appreciation of the ideas and results presented in the paper. We summarize the main questions from the reviewers and address them separately below.

To Reviewer #1 Q1: Network connectivity is presumably known . . . it seems all the graphs considered are complete graphs. We note that the network connectivity is not assumed to be known. Agents only interact with their local neighbors and do not know the entire network structure. There are no constraints on the network connectivity to guarantee the convergence of the proposed algorithm. Moreover, in experiments, we have considered networks that are **not complete** (**Figure 1a** in the paper), which have similar experimental results to that of the complete networks.

Q2: **Include more description about digit classification/some refs . . .** We thank the reviewer for the useful comment and will include further description and refs in the revision.







To Reviewer #2 Q1: Comparison to [5, 16]. We add experimental comparison of our work with [5] and [16] here ([5] and this paper: 30 agents, human action recognition; [16]: 100 agents, target localization). We note that [5] considers fault-tolerance to dropped nodes (that may stop sending message), whereas [16] and this paper consider a more general resilience to Byz. attacks (that can send arbitrary messages). The results show that our method is also resilient to attacks consisting of dropped nodes (Fig. 3). In contrast, [5] fails in the Byz. systems (Fig. 1)—as the number of Byz. agents increases, test loss also increases. Since [16] has the same Byz. setting as this paper, we omit experiments using our method in the setting of [16]. In contrast to our method, [16] requires a user defined parameter F, which is the maximum number of Byz. agents in the neighborhood of a normal agent. If the selected F is smaller than the actual number of Byz. neighbors, then [16] fails (see Fig. 2, actual maximum number of Byz. neighbors is 2, by setting $F = 0$ or 1, [16] results in a worse learning performance/larger MSD compared to no-cooperation). In comparison, **this paper is resilient to an arbitrary number of Byz. agents and does not require the input F**. Besides, the time complexity for [16] is exponential in F, making it infeasible for large networks and large number of Byz. neighbors, whereas **this paper has linear time complexity.**

To Reviewer #3 Q1: Scope of the paper/Missing related work. There is a large body of related work to MTL with different variations. The suggested refs by the reviewer mostly deal with a different aspect in MTL, which is more related to transfer learning and shares a different motivation/assumption compared to this paper (see Fig.4,5). The first MTL setup usually assumes a known relationship between tasks (e.g., learning depth/semantics from RGB images simultaneously since the two share related representations), has data beforehand and learns in a fusion center. It usually learns multiple objectives from a shared representation by sharing layers and splitting architecture in the deep NN, e.g., sharing the first several layers with all the tasks and only the last layers are task-specific. In contrast, we consider a network of agents that maintain separate models without sharing layers, the relationship between agents is unknown, data is not collected centrally and agents learn in a distributed manner. These two MTL setups have different applications: The first is widely used in Deep Learning, e.g., CV and NLP, whereas the second is naturally suited to model distributed learning in **multi-agent systems** such as mobile phones, autonomous vehicles, and smart cities. We also note that the suggested refs. *"An Overview . . . "*, *"MTL using uncertainty . . . "* and *"cross-stitch . . . "* are about sharing layers/architecture of NN, which is not related to our MTL setting; *"Large scale . . . "* and *"FedNAS"* are about distributed learning with deep models but not MTL. We can add an explanation to clarify the MTL scope of the paper.

R. Caruana. Multitask learning.

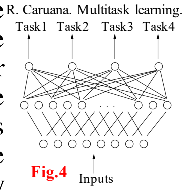


Fig.4

MTL by Reviewer #3

Ali H. Sayed, et. al. Multitask learning over graphs.

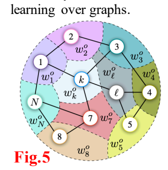
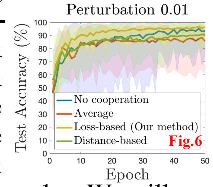


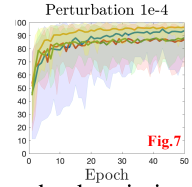
Fig.5

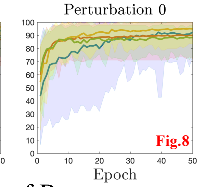
MTL we consider

Q2: Convex model assumption. Convex models are typically assumed in the ML literature for convergence analysis. Although the analysis is based on convex models, we also used non-convex models, such as CNN in digit classification (Table 1), and obtained experimental results that are similar to convex models. For non-convex models, the loss is computed using the same approach as convex models and therefore, no alternative way is needed.

Q3: Experiments related to deep MTL. We have used a CNN for digit classification and compared our method with others in this deep distributed MTL system, and show the superiority of the proposed method.







To Reviewer #4 Q1: Byz. definition/Small perturbation attack. In the analysis, we show the convergence of the algorithm in the presence of Byz. agents sending arbitrary messages. In experiments, the particular messages sent by Byz. agents can be found in Appendix B. Byz. agents send random values from the interval $[15,\ 16]$ (in each dimension) in the target localization example, and from the interval $[0,\ 0.1]$ in the classification examples. We will move the description of Byz. messages to the main context in the revision. We also provide additional experiments for small perturbation examples here (for human action recognition, 30 agents, 10 Byz.). Results are similar to the ones in the manuscript when perturbations are small. We also note that when perturbation is 0, the scenario degenerates to the non-attack case.

Q2: Push derivation to . . . . To improve readability, we will include explanation of the method in the beginning of the derivation and move some of the derivations to the appendix in the revision, as suggested by the reviewer.