Reviewer 1. We agree with **R1** that the two directions mentioned are natural. **(1)** Our **lower bound** is likely to be too loose. Note however that the factor in the second summand of regret for no communication (cf. line 278) is N, *irrespective of the graph structure*. A question would be what the second summand can be if the first one is proportional to $\log(TN)$. We have not found yet a way to translate the impossibility of mixing information arbitrarily quickly to a lower bound on the regret. It is tricky given the statistical properties of our problem, in which as time grows, neither mixing nor the graph should be that important. At the same time, establishing *graph-dependent lower bounds for decentralized methods is an active area of research in its own right*. Despite the vast literature on decentralized convex optimization, for instance, the first graph-dependent lower bounds appeared only recently in the NeurIPS 2018 paper [30]. **(2)** One would also expect that as time grows *less communication should work well enough*. We have explored this path (counting the number of mixing stages that would be needed) and have found it is not straightforward, although we know that this quantity at least has to be linear in N (stages). What plays an important role is that most ways that increase delay (equivalently, reduce communication) introduce a multiplicative factor depending on N to the $\log(TN)$ regret term, which is inadmissible, since to match the same regret T would need an exponential increase. This is because more delay is similar to having a different ε (line 571).

Reviewer 2. New techniques, new (not incremental) results and importance of acceleration. *Our work focuses on the dependence of the regret as a function of the network size, N, not just as a function of the time horizon, T.* In this respect, **our results on the regret are the first of their kind and are not incremental**: **(1)** We have the first decentralized algorithm with a $\log(TN)$ regret term with no factors dependent of the network size and topology. The factor is important because in a comparison, our T would need to be exponentially larger to yield regret as large as the ones where a poly N factor appears in front of the $\log T$ term, as in Landgren et al. **(2) Accelerated method does not yield same regret**. Acceleration gives a square root improvement in the denominator of the second summand, which improves regret w.r.t. N (cf. lines $574-576$). **(3)** Take the example in Appendix C, for instance: when evaluated it yields a factor in the second summand of regret of $\Theta(N^3 \log N)$ (unacc.) instead of $\Theta(N^2 \log N)$ (acc.), which is compared to the worse rate $\Theta(N^{7/2})$ in Landgren et al. This makes a **crucial difference in large networks**. **(4)** About this example, note that the matrix **R2** proposes is only making the process lazy and thus the mixing strictly slower. The new matrix amounts to $\frac{1}{3}I + \frac{2}{3}P$, and the same computations in our paper yield $\lambda_i = \frac{1}{3} + \frac{2}{3}\cos(2(i-1)\pi/N)$, $\frac{\lambda_i^2}{1-\lambda_i^2} \geq \frac{3N^2}{8(i-1)^2\pi^2} - \frac{5}{8}$, which is of the same order as the matrix P we consider, so the exact same asymptotics apply. **(5)** One cannot compare directly the performance of the acc. and unacc. methods in the experiments, since the delay in the unacc. one was picked to be the delay predicted for the *acc. theorem*, as mentioned in the section and indicated in the plot. The acc. one would probably improve if ε is tuned too. Our theory predicts a gap between the unacc. and acc. methods that does not depend on T and that increases when N grows (i.e. λ_2 increases). This gap is not constant (it depends on N) and it can be seen in the experiments. **(6)** A lot of effort has been put in finding optimal algorithms for distributed systems, and **acceleration techniques are key to achieve optimality**. See for instance [30], which won one of the best paper awards at NeurIPS 2018, and that could find optimality in their setting under local regularity via a Chebyshev acceleration argument as well. See **R3**'s section on global structures.

Other differences compared to Landgren et al.: (7) We use delays, which give **lower variance estimators**. **(8)** We **use less information**, which is of interest for **decentralization purposes** and has important **computational implications**. As we write in the paper, the algorithm of Landgren et al. requires the full set of eigenvalues and eigenvectors of P to compute ε_c. This requires full knowledge of P and $O(N^3)$ operations. In our case, agents do not need to know the entire matrix P (only the spectral gap of P, as typical in decentralized methods [30]) and our comp. cost is constant per iteration and node. Just computing ε_c in Landgren et al. for N larger than 200 takes several hours. The two algorithms are very different. *We believe our algorithm gives lots of new insights for the understanding of decentralized bandits.*

Reviewer 3. The paper focuses on the statistical problem of minimizing regret, something of interest in its own right. Analyzing communication cost is beyond the scope of this paper (see response to **R1.(2)**). The assumptions of the distributed system are included in Sec 2 and they correspond to the standard ones typically considered in the literature on decentralized methods, cf. [30, 31]. Our main focus was theoretical work and we added the experiments as proof of concept rather than an aiming at an exhaustive study. We provided theorems and compared the regrets obtained, the experiments show that the theoretical results carry on in practice. The matrix P is a classical object in decentralized methods. Only the maximum degree of the graph is needed to construct P [31]. We will add details in the final submission. There are no identifiers. In the introduction of the paper that R3 cited one can read that averages are feasible, that is the only thing we use. The comparison of various algorithms is fair, all of them use the same communication. **Decentralization (without global structures and broadcasting) is important**, as reflected by the vast and growing literature in machine learning. **(1)** It is a first step to address time varying graphs or faulty networks (cf. 61), and in systems with privacy/communication constraints. **(2)** It has many applications, like sensor networks (cf. 71 & [31]). The topology is grid-like, agents can only interact with close neighbors, and it is very different from complete graphs where broadcasting can be implemented. We will add a comment on this.