We thank the reviewers for their thoughtful feedback, as well as the pointers to related work. We will address a few of the major comments below, and will incorporate these points and discussions into the revised paper.

First and foremost, we want to clarify some misunderstandings in Reviewer 2's comments on the paper, specifically the statements that "the algorithms is still too close to local search and the only gain from the approach is the initialization of the local search", and that "a fair comparison could be to give many random initializations for the local search approach as can be afforded in the same time as the approach in this paper."

We want to emphasize that this comparison is precisely what we did in the evaluation (we realize this can be stated more clearly and we will absolutely edit the paper to emphasize this point). Specifically, for the results in Table 2, and as stated in lines 263–268 of the text, the results we report are for Louvain/Leiden with 10 random restarts (the best way to improve their performance given the additional time), whereas the Locale entries use only a single initialization. Thus, the time taken for Louvain/Leiden in that comparison is *more* than the time using the Locale initialization, and Locale still performs better. While it is true that we also compare to semidefinite solvers, the above comparison is the practical point of emphasis when it comes to the speed of the method. We agree that the "10 iterations" (which counts the number of computing embeddings, rounding, and refining) vs. "10 trials" (which counts the number of random restarts) is a bit confusing upon re-reading, and we will absolutely clarify this in the resubmission, plus explicitly show time/performance curves for the different methods. We hope that this addresses Reviewer 2's main concern with the method, ensuring that we are indeed conducting an apples to apples comparison, where the Locale approach brings substantial benefits.

We next want to address several other comments made by the reviewers.

R2 and R4 *(on including pseudo-code of the mentioned algorithm)*: Thanks for the suggestion. We agree that this will help to make the paper more self-contained and can certainly add this.

R3 *(on comparisons to other community detection approaches)*: You are absolutely correct that the method and analysis here definitely do focus on improving the modularity optimization of Louvain/Leiden. While we completely agree that the real goal is the general community detection, and modularity is a single approach for this task, the methods are popular and ubiquitous enough that we believe a substantial improvement on them to be an important contribution to the literature (and so we focus largely on this improvement). However, we're definitely happy to provide additional experiments on communities with ground truth data from SNAP with a brief comparison to higher-order methods.

R4 *(on the discussion about k)*: Sorry for the confusion on this point. We do not assume that we know the number of clusters overall (denoted by symbol r). Instead, the symbol k in section 3 denotes the cardinality of candidate memberships for *each vertex* (i.e., this would be $k = 1$ for the greedy local move of Louvain/Leiden, even though they can of course discover many clusters). We observe that a small k is sufficient to reach the global optimum of the SDP, but larger k is also fine. However, since the Locale algorithm has a time complexity of $O(\mathrm{card}(A)klogk)$, we prefer a smaller k as long as it is sufficient to satisfactorily optimize the SDP. We will clarify this in the paper.

R4 *(on Fig 2)*: Yes, the optimal value in Figure 2 means the modularity score. We appreciate the suggestion and will change it to the relative ratio.

R4 *(on additional references mentioned in the review)*: Thanks for the references! We will absolutely discuss them in the paper.

R2, R3, and R4: We appreciate the feedback on writing and presentation and will incorporate these suggestions into the paper.