We thank all reviewers for their detailed and insightful reports. As pointed out by most reviewers, the key to our work is that, by using complex-valued matrices, the k clusters in a digraph can be well separated by *angles* in $\mathbb{R}^2$ with the help of a *single* eigenvector. The result in Peng et al. [22] only holds for undirected graphs and k eigenvectors are needed to construct an embedding. Generalising their result into the setting of digraphs is a mathematically involved task.

Reply to R1: We thank R1 for recognising the conceptual contribution of introducing the flow ratio. For the comment on pairwise different clusters not being captured in the objective function, notice that if one takes all the pairwise clusters into account, the objective function would involve $\Theta(k^2)$ terms. Even if the remaining $\Theta(k^2)$ terms are much smaller than the ones along the path, their sum could still be dominant, leaving little valuable information on the cluster structure for the objective function. Instead, we show that when the relationship between clusters present a structure, one should *only* take the cut values along the path into account. This difference makes it difficult to compare our objective function with others. The motivation for studying a flow ratio is witnessed by our experiments on the international trade data set and South Korean COVID-19 dataset, on which meaningful clusters are found. On the problem's hardness, notice that, when $k = 2$, the optimal solution is the bipartition of V that maximises $w(S_0, S_1)$. This is exactly the MAX DICUT problem, which is NP-hard (cf. Goemans and Williamson, JACM'95). We will add this in the final version.

For the DSBM experiments, the sizes of the input graphs are $n = 1000$ for Fig.1, and $n = 2000$ for Fig.2. We accidentally omitted these, but will report these values in the final version. The number of vertices n and clusters k we choose are in line with many references for (D)SBM experiments, and we do remark that the results of different runs are very close. Hence, we believe that the number of runs needed for good convergence should not play a significant role here so that using 5 runs is sufficient. We follow reference [9] and use the ARI value for evaluating the clustering quality. We will consider reporting other metrics such as the number of misclassified vertices, precision or recall. The inner and outer conductance are not suitable for our setting since cluster structures in Fig.1 are characterised by edge directions instead of densities. R1 is correct that the term in L.185 should be $\mathsf{APT}/(\gamma_k(G) - 1)$, and this will be corrected.

Reply to R2: We thank the reviewer for highlighting both the new theoretical and experimental contributions of the paper, and appreciate their comments on the paper exposition. The page limit restricted extended discussion, but we will improve the presentation. Regarding Eq.(3), we did not add more discussions/references since both (3) and the equation before are new. In particular, the equation before (3) already uses angles to differentiate clusters, so it is not a standard normalised indicator vector as used for undirected graphs. Because both the equations are conceptually new, we did not add more discussions here. Instead, L.110-115 compared these two equations with previous work.

Regarding the detailed comments: For L.42, we do mention that "for any set of vertices $S_0, \ldots, S_{k-1}$ *that forms a partition*"; For L.43-45, we will move the definition of $w(\cdot, \cdot)$ and $\mathrm{vol}(\cdot)$ up; For L.73-74, we will drop the second equality; For L.98-98, we will rewrite it as "which cross the cut from S_j *to* S_{j-1}"; For L.138-139, here we use f_1 as an embedding, and every vertex v maps to $f_1(v) \in \mathbb{C}$; these n embedded points are the input of k-means: we will make it more clear; On the comment about S4.2, the three lemmas are used as key components to prove the theorem. Specifically, we prove by contradiction through a quite technical analysis that if the theorem does not hold, then one of the lemmas does not hold. We will add some discussion about this in the final version. Now we address R2's concern on L.112 (and Correctness). Notice that the denominator is equal to 0 if $\sum_{j=1}^{k-1} w(S_j, S_{j-1})/k(\mathrm{vol}(S_j) + \mathrm{vol}(S_{j-1})) = 1/4$. This summation is maximised if the graph is complete k-partite and every partition is a cluster. Moreover, the case in which θ_k achieves maximum occurs when $k = 2$, which leads to $1 - (4/k) \cdot \theta_k = 0$. This is indeed a corner case, we will clarify this (or adjust constants in the denominator of $\gamma_k(G)$ to avoid this). We thank the reviewer for pointing this out.

Reply to R3: We thank the reviewer for their thoughtful comments. Regarding the comment on comparing our method with the ones for partitioning undirected graphs, notice that our key contribution is to introduce the flow ratio, and demonstrate that, thanks to complex-valued matrices, the cluster structure can be encoded into *a single eigenvector* (instead of k eigenvectors); different clusters are then separated by *angles* in the $\mathbb{R}^2$ embedding. R1 highlighted this as a strength. For the experiments, the DSBM graphs have 1K to 2K vertices, similar in size to other references. We also evaluated our algorithm on the UN Comtrade Dataset (zipped 99.8GB). Therefore, we do believe our algorithm has been tested on large and relevant graph instances.

Regarding R3's comment on comparison to PageRank and matrix factorisation methods, we specifically compared our algorithm with all the previous ones addressing the same goal. To the best of our knowledge, there is no PageRank-based algorithm to find clusters in digraphs whose structure are defined with respect to flow imbalance; furthermore, all matrix multiplication based algorithms have a runtime of $\Omega(n^\omega)$, which is computationally more expensive than our algorithm.

Regarding the comments on the assumption on k, notice that for most applications k is small and it is reasonable to focus on $k = O(\mathrm{poly} \log n)$. The use of $\tilde{O}$-notation to hide $\mathrm{poly} \log n$ factors is a standard practice in studying nearly-linear time graph algorithms. About the pseudocode: because of the simplicity of the algorithm, whose entire description is included in L.137-142 and L.154-164, we did not include the pseudocode due to space limitations. Should R3 strongly believe that inclusion of pseudocode would significantly improve the readability of the paper, we can aim to do so.