We thank the reviewers for their thorough reviews and useful suggestions. We address below the main concerns raised by the reviewers. These will be incorporated in the final version, along with corrections of minor comments.

Technical aspects of the algorithm. (R1, R2) We chose the parameters τ (Sinkhorn) and γ (learning rate) with grid search prior to running experiments, while S (sampling size) was fixed empirically. In all experiments, we set $\tau = 5, \gamma = 0.2$, and $S = 30$. We set the maximal number of Sinkhorn iterations to 10, and we run stochastic gradient descent for 3000 iterations (even though the algorithm converges long before, after around 1000 iterations, typically). Furthermore, gradient descent was performed in PyTorch with AMSGrad (Reddi *et al.*, ICLR 2018), using automatic differentiation. We did not run the algorithm multiple times to avoid local minima, as the stochastic exploration empirically proved to be very successful in the task. Regarding the sensitivity to different initialisation, the algorithm seems robust in our experiments, and we used random initialisation in all our experiments. Finally, we bypass the computation of the pseudo-inverses by adding a small diagonal shift to the Laplacian matrices and directly computing the inverse matrices, which is orders of magnitude faster. We will add this information to the final version of the paper.

Convergence of the algorithm and computational complexity. (R1, R2, R3) Under mild assumptions, GOT algorithm converges almost surely to a local minimum. We do not have any guarantees regarding the global minimum, as the problem is nonconvex. In practice, we observed that the algorithm fails to converge to a good solution only when the compared graphs are very different (we expect to solve this issue with many-to-many correspondences, see next answer). As for the computational complexity, our unoptimized implementation is $O(N^3)$ per iteration. We are currently looking into ways to improve the complexity, by exploiting the sparsity of matrices, and by using fast algorithms to compute the matrix square-root gradient (Lin *et al.*, BMVC 2017).

Graphs of different sizes (R1, R2). We will make it clearer in text that our algorithm is aimed at graphs of the same size. In this regard, please note that our method can be extended to graphs of different sizes. We are currently working on the idea of allowing many-to-many correspondences between vertices, which generalises the one-to-one matching considered in our current approach. This will also make it more robust for comparing graphs that cannot be matched perfectly. Given the significant body of additional material, we feel that this topic is best left to a future publication.

The distance captures the "global structure of graphs" (R1) Although we do not have a formal proof due to the lack of standard definition for the global structure, our main idea is that graphs can be seen through the distribution of smooth signals. Therefore, the discrepancy in lower graph frequencies, known to capture the global graph structure, will be more important in defining the distance. We supported this claim with the example in Figure 1, where two perturbed copies are compared to the original graph, both with the same amount of removed edges, but a different degree of structural change. The example shows that the proposed distance manages to capture the structural change efficiently.

Missing references (R2) We thank the reviewer for suggesting several important works to complement the graph alignment overview. We will gladly include them in the final version. It will be interesting to see whether any of the proposed methods can improve the results of our algorithm, and we will make sure to investigate this in the future. **For directed graphs (R2)** The problem in extending this work to directed graphs indeed lies in the definition of the Laplacian (which is not unique, and it is not positive semi-definite, with possible complex eigenvalues). We do, however, agree it would be very interesting to explore possible extensions of our work to directed graphs.

Behaviour of the optimal transportation plan (R3) While it is difficult to derive any guarantees on the behaviour of the transport plan even if the observed signals are smooth, we note that the transport plan is a continuous Lipschitz bounded operator, suggesting some regularity in its behaviour for all signals. That being said, we do agree that signals which are more likely in the observed distribution will have a more robust transportation. An interesting direction would be to derive a transportation plan through a graph filter, a more appropriate Gaussian model for non-smooth signals (Segarra *et al.*, TSIPN 2016). **Use of KL divergence (R3)** In our experiments with KL divergence, numerical issues prevented the algorithm to run until convergence. However, we did not investigate the question thoroughly, and we will rephrase the sentence on line 136-137 to avoid any confusion. Nonetheless, it would be interesting to compare this approach to our method. **Example of Graph Signal Transportation (R3)** The nearest neighbours are computed with the Euclidean distance. The edge weights in the constructed k-NN graph are all equal to 1. The Laplacian matrix is then squared (multiplied by itself) to capture 2-hop distances and create more meaningful weights. The number of neighbours was chosen arbitrarily. We note here that the construction of the graphs is important and a more meaningful graph (obtained through graph inference methods for example) could potentially yield better results. **Comparisons with NetLSD (R3).** We agree that NetLSD is a very interesting method with a different approach that could be used for graph classification. Our preliminary results on the experiment in Figure 5 indicates that it yields 74/100 accuracy. While not as good as some other investigated methods, we note that this is a very good score for a method not taking into account any sort of alignment of nodes. We will add the full comparison to NetLSD in the final version of the paper, including the confusion matrix.