We thank the reviewers for the valuable comments, which require simple changes to the manuscript.

R1: 1) We can easily address the comments on the writing, while pushing some unnecessary technical details to the supplementary material. We remark that we had a typo in line 86 and should instead have $\sigma^*_{ij} = \sigma^{*-1}_{ji}$. 2) We will include the mentioned references. 3a) Condition (12) in our model can be interpreted as follows: in a 3-cycle, corrupting two edges induces on average more cycle-inconsistency than corrupting a single edge. This is currently explained in line 503 of the supp. material (also note our correction in line 614). Corruption with a uniform (Haar) distribution also satisfies this condition (with equality), however, it has a very strong and unrealistic assumption on the distribution. For simplicity we have a star-shaped topology of our graph, though with more work (and rather complicated descriptions) one may generalize the topology of our graph. We emphasize departure from the uniformity assumption on the distribution of corruption and not the "uniform topology" of the graph. 3b) Although our theory and experiments assume a complete graph, for simplicity of demonstration, they can be easily generalized to the incomplete Erdős-Rényi graphs. We will add this to the supp. material. 4) We remark that IRGCL outperforms CEMP+weighted least squres (WLS), which reports $\boldsymbol{P}_{(1)}$, in highly corrupted scenarios. For example, under the uniform corruption model with $n = 100$ and 90% corrupted edges, the error of IRGCL is < 0.1, while $\boldsymbol{P}_{(1)}$ gives estimation error > 0.5 (like other methods reported in Figure 3). That is, given poor initialization, IRGCL can still converge to a reasonable solution. We believe that IRGCL uses additional cycle consistency information to adjust the edge weights (see also reply to R3). However, under mild corruption, the CEMP initialization is accurate and helpful in accelerating the convergence and this is what the theory verifies (previous theory of [17] for high uniform corruption requires very large n). 5) We will report the result of CEMP+WLS (that is, using $\boldsymbol{P}_{(1)}$) in the updated version. 6) The original real data used in our work is the most challenging one for permutation synchronization. However, it contains many nodes whose neighboring edges are completely corrupted. In such a case, none of the permutation synchronization algorithms work well and additional information, such as coordinates of key points, is needed. Thus, in order to make a valid evaluation of different algorithms, we have to preprocess this data so that permutation synchronization is well-posed. We remark that in many SfM data (e.g. the initial matching used in [20]) such a malicious scenario does not occur (thus no such preprocessing is needed); however, currently our algorithm and other direct algorithms for permutation synchronization cannot be easily applied to SfM data since they deal with permutations and not partial ones. The nontrivial extension to partial permutations is left for future work.

R2: 1). We will demonstrate this in practice. In theory, for 3-cycles and uniform corruption, a necessary condition for CEMP and IRGCL is $p = \Omega(1/\sqrt{n})$. According to [17], up to a log factor this condition is sufficient for CEMP. As the length of the cycle increases the lower bound on p decreases and approaches $p = \Omega(\log n/n)$ as the size of the cycle approaches infinity. The complexity of our reformulation of CEMP is at most n^3 times the length of the cycle, so in practice we cannot achieve the $\log n/n$ threshold of disconnectivity, but get close to it. 2) We will report runtimes in the new version. We remark that IRGCL is often slightly slower than IRLS, but they are comparable to each other. Our experiments indicate the following order of runtimes: Spectral<PPM<IRLS<IRGCL≪MatchALS<MatchLift.

R3: The challenge of permutation synchronization is not just its nonconvexity, but more importantly, its discrete and combinatorial nature. IRLS has been carefully studied and tested in some continuous settings, but in discrete settings IRLS is neither commonly applied nor studied. Indeed, in lines 152-160, we explain the drawback of IRLS in our discrete setting. IRGCL handles the limitations of IRLS in the following ways. 1) Recall that IRLS first locally estimates a residual based on a single measurement of the corresponding edge. However, IRGCL estimates the residual using more global information of other edges (reflected in the powers of the GCW matrix). Thus its WLS is much less dependent on the initialization than standard IRLS. Point 4) in our reply to R1 confirms that in practice IRGCL can handle cases where CEMP provides a bad initialization. 2) The solution of the IRLS problem uses convex relaxation of the nonconvex WLS problem. While IRGCL also follows such a scheme, it also uses 3-cycle consistency information which helps more faithfully recover the underlying corruption and thus provide more accurate weights and consequently a better approximation by the convex relaxation. 3) Our edge weights are computed as a weighted average (ideally, expectation) of the 3-cycle consistency (encoded in the square of the GCW matrix). This expectation lies in a continuous space (as oppose to the weights in IRLS that lie in a discrete space). Thus our reweighting scheme smooths the space of edge weights, making the algorithm less likely to get stuck. We remark that CEMP does not explicitly minimize an objective function, but it aims to find the underlying maximal cycle-consistent subgraph (see page 10 of [17]). It can also be interpreted as an iterative procedure that aims to estimate the expectation of the corruption level of edges (latent variables) given their posterior distribution, which is similar to EM. However, its iterations do not rely on MLE and are more efficient and thus have some nice guarantees of convergence (see [17]). Similarly, IRGCL aims to solve a WLS problem whose weights focus on the underlying globally cycle-consistent subgraph. In order to do this, it estimates both permutations (using WLS) and the cycle-consistent subgraph (using CEMP-like reweighting) in an alternating manner.

R4: This review is an outlier in terms of short length, tone, clarity and score. As explained to R3, our work has some nonstandard ideas and we strived to make it accessible, while avoiding some of the complicated ideas of [17] (and it is nice that we have a direct formulation by the connection graph). We will follow all the constructive suggestions and easily improve the clarity, but we disagree that we did not organize our thoughts. The answer to all your points are above, except for the following: 1) The graph is not "a singly connected cycle graph" (and we never stated this). 2) Most previous works on permutation synchronization use real datasets whose images have similar views and share the same set of keypoints (so that keypoint matches are permutations) and their graphs are thus complete. Applying permutation synchronization algorithms to a set of images with distinct views is unrealistic and is not a common practice (the reviewer may be confused with rotation synchronization). Future work will try to generalize our ideas to partial permutations so that we may explore more general SfM data.