# Efficient Detection of Long Consistent Cycles and its Application to Distributed Synchronization *

Shaohan Li
University of Minnesota
li000743@umn.edu

Yunpeng Shi
University of California, Davis
ypshi@ucdavis.edu

Gilad Lerman
University of Minnesota
lerman@umn.edu

2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) | 979-8-3503-5300-6/24/$31.00 ©2024 IEEE | DOI: 10.1109/CVPR52733.2024.00503

## Abstract

*Group synchronization plays a crucial role in global pipelines for Structure from Motion (SfM). Its formulation is nonconvex and it is faced with highly corrupted measurements. Cycle consistency has been effective in addressing these challenges. However, computationally efficient solutions are needed for cycles longer than three, especially in practical scenarios where 3-cycles are unavailable. To overcome this computational bottleneck, we propose an algorithm for group synchronization that leverages information from cycles of lengths ranging from three to six with a time complexity of order $O(n^3)$ (or $O(n^{2.373})$ when using a faster matrix multiplication algorithm). We establish non-trivial theory for this and related methods that achieves competitive sample complexity, assuming the uniform corruption model. To advocate the practical need for our method, we consider distributed group synchronization, which requires at least 4-cycles, and we illustrate state-of-the-art performance by our method in this context.*

## 1. Introduction

Structure from Motion (SfM) asks to recover the 3D structure of a stationary scene from multiple images taken by cameras from different orientations and locations. In the past decade, the global SfM pipeline has become increasingly popular due to its several advantages over the incremental pipelines [17, 31]. First of all, global SfM requires only one implementation of bundle adjustment, making it more efficient in computation. Second, it estimates camera poses in a global optimization framework which mitigates the drifting issue of the incremental pipelines. Despite the popularity of global SfM pipelines, the estimation of global camera poses (e.g., orientations) remains a highly challenging problem. For instance, estimating camera orientations from their relative measurements, often called rotation synchronization, is a highly nonconvex graph optimization problem. In typical scenarios of highly noisy or corrupted measurements of relative orientations, many common solutions of rotation synchronization have poor accuracy and slow convergence.

Given these challenges, theoretical developments have demonstrated the critical role of cycle-consistency information in inferring corrupted measurements [25]. In practice, the consistency constraint on 3-cycles was utilized to estimate the error of each measured relative orientation. It also helped nonconvex iterative rotation synchronization solvers avoid spurious local minima and achieve significantly higher accuracy [39]. However, the usage of 3-cycles largely limits the application of these improved algorithms to other important scenarios. One scenario involves a sparse viewing graph lacking sufficient 3-cycles. This often occurs when the size of the graph is too large to densely measure the relative orientations on its edges, which could happen in certain case for the molecular orientation estimation in cryo-electron microscopy imaging. Another scenario is orientation estimation for each piece of jigsaw puzzles, where the graph is a 2D lattice and 3-cycle does not exist. Lastly, in distributed SfM, edges between any two clusters of nodes form a bipartite graph, and cycles of odd length do not exist. Our numerical results primarily emphasize the practical scenario of distributed SfM, which holds particular relevance for the broader computer vision community.

Despite the multiple critical applications of long-cycle consistency, inferring measurement noise from long cycles is challenging in both computation and theory. First of all, the number of cycles grows exponentially with the cycle length, and measuring cycle inconsistencies for each long cycle is computationally intractable. Moreover, developing theoretical guarantees for long-cycle inference methods is fundamentally more difficult than the 3-cycle case. Indeed, in a random graph setting, a set of longer cycles are more likely to share common edges, making their consistency score highly correlated. Therefore, new tools are required to handle the correlated empirical process.

In this work, we propose the first practical method, LongSync, for inferring edge corruption levels from long cycle consistency information. For this purpose, we carefully modify and vectorize the Cycle Edge Message Passing (CEMP) method [25]. This nontrivial modification drastically reduces its computational complexity when using longer cycles. Moreover,

*This work was supported by NSF award DMS-2152766.
Supplementary code: https://github.com/sli743/LongSync

2575-7075/24/$31.00 ©2024 IEEE
DOI 10.1109/CVPR52733.2024.00503

Authorized licensed use limited to: University Of Minnesota Duluth. Downloaded on July 29,2025 at 06:11:35 UTC from IEEE Xplore. Restrictions apply.

by employing a more delicate analysis and incorporating new tools from probability theory and combinatorics, we develop a significantly stronger exact recovery result with a general cycle length under a popular probabilistic model. The sample complexity in our theory is the lowest among all practical rotation synchronization methods. Although we limit our scope to the application of distributed SfM, our algorithm and theory applies to any finite-dimensional linear group in group synchronization, and not just $SO(3)$ in rotation synchronization.

## 1.1. Related Work

Earlier rotation synchronization methods [2, 10–13, 18, 33, 42, 43] seek to minimize a least squares energy function. They can be described as relaxed versions of the maximum likelihood estimator under the additive Gaussian noise model, but they are not robust in the presence of outliers or heavy-tailed noise. Nevertheless, in the case of global SfM, the initially estimated relative camera rotations can be severely corrupted due to the erroneous keypoint matches and the subsequent poor estimation of fundamental matrices.

To handle outliers, robust rotation synchronization methods either minimize a robust energy function or reweigh/trim the viewing graph based on the corruption levels of the edges. Wang and Singer [49] minimizes a corresponding $\ell_1$ objective function using semidefinite programming (SDP) relaxation, which is slow in practice. Other energy minimization methods are typically nonconvex, which include the Weiszfeld algorithm [20] and the Riemannian subgradient method [29] for $\ell_1$ minimization, and the iteratively reweighted least squares (IRLS) for minimizing general $\ell_p$ [5] and Geman-McClure [4] loss functions. However, all these methods heavily rely on good initialization. Sidhartha and Govindu [41] partially remedy the issue using adaptive Geman-McClure loss functions, but their approach remains sensitive to the initialized weights. Maunu and Lerman [30] propose to solve rotation synchronization by an iterative robust averaging method that utilizes Tukey depth, but they have not demonstrated effective performance for real SfM applications. Arrigoni et al. [1] applies a low-rank and sparse matrix decomposition method to SO(3) and SE(3) synchronization, but it is even less robust to outliers than IRLS.

Instead of employing a robust objective function, Shen et al. [36] and Zach et al. [51] uses the 3-cycle consistency constraint to detect and remove corrupted relative orientations. Lerman and Shi [25] take one step further to estimate the corruption level of each relative measurement by a novel cycle-edge message passing (CEMP) algorithm. They then use the estimated corruption levels to reweigh the graph and solve rotation synchronization using a weighted least squares method. This message passing procedure was further combined with IRLS to boost its accuracy in [39]. Particular versions and extensions of this procedure for permutation and partial permutation synchronization, which are relevant to the matching component of SfM, were discussed in [27, 40].

However, all the previously mentioned cycle-based methods [25, 27, 36, 38–40, 51] only use 3-cycles in practice, limiting their application for distributed synchronization. Indeed, the standard distributed synchronization often requires "stitching" local solutions by synchronizing the relative rotations between clusters. Each of these inter-cluster rotations is estimated by "averaging" the edges between the two clusters. These edges form a bipartite graph, and the minimal cycle length is 4. As pointed in [51], the number of operations for computing long cycle consistency information scales exponentially with the cycle length. Therefore, none of the existing distributed rotation synchronization methods directly exploits long cycle information due to this computational challenge.

The earlier distributed methods for $SO(d)$ synchronization, such as [45] and [44], minimize a least squares energy and are not robust to outliers. A series of distributed SfM methods [14–16] implement incremental SfM algorithms for each cluster. However, these methods do not employ a standard rotation synchronization algorithm, as they require additional information such as the number of keypoint matches between images. Moreover, the incremental methods are slower since they require multiple rounds of global rotation synchronization. MultiSync [9] synchronizes the inter-cluster rotations directly using all inter-cluster edges among all clusters, by formulating a novel synchronization problem on a multi-graph. Although it utilizes a more unified formulation, its objective function is least squares which largely limits its robustness to outliers.

A recent and different type of methods for rotation synchronization use deep learning [21, 26, 34]. However, these methods are supervised and thus may not generalize well when switching datasets. Moreover, like many other previous methods, they lack theoretical guarantees.

A common theoretical setting to assess the performance of rotation synchronization algorithms is the uniform corruption model (UCM) described in §4. We provide the best sample complexity for LongSync, even with only 3 cycles, among all previously established estimates for the UCM model.

## 1.2. Contributions of This Work

- We propose the first practical algorithm that infers edge corruption levels from long cycle consistency information. The computation complexity of our method is reduced from $O(n^c)$ to $O(n^3)$ (or possibly $O(n^{2.373})$) for cycle length $c \leq 6$ and $O(n^{\lceil (c+3)/2 \rceil})$ for $c > 6$.
- We establish sample complexity estimates for our method under the uniform corruption model, where we get closer to the information theoretic bound than any other existing work. Our proof requires delicate analysis and it also improves previous estimates for the CEMP algorithm.
- We introduce a new graph partition and graph preprocessing method that utilizes our inference method, and demonstrate the effectiveness of our pipeline in boosting the performance of distributed synchronization.



Authorized licensed use limited to: University Of Minnesota Duluth. Downloaded on July 29,2025 at 06:11:35 UTC from IEEE Xplore. Restrictions apply.

- Extensive numerical experiments demonstrate the outstanding performance of our method.

## 2. Problem Formulation and Preliminaries

Assume a graph $G=([n],E)$ where $[n]$ is the set of nodes indexed by $\{1,2,\cdots,n\}$ and $E$ is the set of edges. Given a mathematical group $\mathcal{G}$, each graph node is assigned an underlying ground truth group element $\boldsymbol{R}_i^*$, where $\boldsymbol{R}_i^* \in \mathcal{G}$ and we use star superscript to emphasize the ground truth. For each edge $ij \in E$, we observe a relative group ratio $\boldsymbol{R}_{ij} \in \mathcal{G}$, whose clean counterpart is $\boldsymbol{R}_{ij}^* = \boldsymbol{R}_i^* \boldsymbol{R}_j^{*-1}$. **Group synchronization** aims to recover the ground truth group elements $\{\boldsymbol{R}_i^*\}_{i\in[n]}$ from the possibly noisy and corrupted measurements $\{\boldsymbol{R}_{ij}\}_{ij\in E}$. In this paper, we focus on the case of rotation synchronization, which is a special case of group synchronization with $\mathcal{G}=$SO($d$). For applications in camera orientation synchronization ($d=3$), we estimate absolute rotations for each node $i\in[n]$ from measured relative rotations of edges in $E$. Note that since $\{\boldsymbol{R}_i^*\}_{i\in[n]}$ and $\{\boldsymbol{R}_i^*\boldsymbol{R}_0\}_{i\in[n]}$ generate the same set of relative rotations, one can only estimate $\{\boldsymbol{R}_i^*\}_{i\in[n]}$ up to a global rotation. The generalization to any linear groups is discussed in the supplementary material.

### 2.1. Notations and Definitions

We denote the adjacency matrix of graph $G$ as $\boldsymbol{A}$, and form a pairwise observation matrix $\boldsymbol{R}\in\mathbb{R}^{dn\times dn}$ by stacking the $\boldsymbol{R}_{ij}$'s (for $ij\notin E$, set $\boldsymbol{R}_{ij}=\boldsymbol{0}_{3\times 3}$):

$$\boldsymbol{R} := \begin{pmatrix} \boldsymbol{R}_{11} & \boldsymbol{R}_{12} & \cdots & \boldsymbol{R}_{1n} \\ \boldsymbol{R}_{21} & \boldsymbol{R}_{22} & \cdots & \boldsymbol{R}_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{R}_{n1} & \boldsymbol{R}_{n2} & \cdots & \boldsymbol{R}_{nn} \end{pmatrix}.$$

We list the matrix operations used in the paper. For matrices $\boldsymbol{X}$ and $\boldsymbol{Y}$, the operations $\boldsymbol{X}\otimes\boldsymbol{Y}$ , $\boldsymbol{X}\odot\boldsymbol{Y}$, $\boldsymbol{X}\oslash\boldsymbol{Y}$ respectively denotes the Kronecker product, Hardmard (element-wise) multiplication and Harmard division between $\boldsymbol{X}$ and $\boldsymbol{Y}$. $\boldsymbol{X}^{\odot k}$ denotes the element-wise matrix $k$-power. For block matrices, $\langle \boldsymbol{X},\boldsymbol{Y}\rangle_{\text{block}}$ denotes the blockwise inner product of $\boldsymbol{X}$ and $\boldsymbol{Y}$, i.e. $\langle \boldsymbol{X},\boldsymbol{Y}\rangle_{\text{block}}(i,j)=\langle \boldsymbol{X}[i,j],\boldsymbol{Y}[i,j]\rangle$, where $[i,j]$ refers to the corresponding block of the matrix.

### 2.2. Review of CEMP for $c$-Cycles

We assume the above setting of $SO(d)$ synchronization. Let $\mathcal{D}$ be any bi-invariant metric on $SO(d)$. We assume a fixed number of cycles, $c$, and denote by $N_{ij}^c$ the set of simple cycles of length $c$ (or simple $c$-cycles) containing edge $ij$. CEMP [25] aims to estimate for each edge $ij$ the corruption level

$$s_{ij}^* = \mathcal{D}(\boldsymbol{R}_{ij},\boldsymbol{R}_{ij}^*), \tag{1}$$

from the set of cycle inconsistency measures

$$d_L = \mathcal{D}(\boldsymbol{R}_L,\boldsymbol{R}_{ij}) \tag{2}$$

where cycle $L=(ik_1, k_1k_2, \cdots, k_{c-2}j, ji) \in N_{ij}^c$ and $\boldsymbol{R}_L := \boldsymbol{R}_{ik_1}\boldsymbol{R}_{k_1k_2}\cdots\boldsymbol{R}_{k_{c-2}j}$. The estimated $s_{ij}^*$ can then be used for extracting a clean subgraph, or to implement a weighted least squares solver where higher weights are assigned to cleaner edges.

It is obvious that if all the edges in $L$ are clean then $d_L=0$. Moreover, due to bi-invariance of $\mathcal{D}$, the following holds true

$$d_L = s_{ij}^* \text{ whenever } L\in G_{ij}^c, \tag{3}$$

where $G_{ij}^c$ is the set of good $c$-cycles with respect to $ij$, i.e. the set of cycles $L\in N_{ij}^c$ such that $ik_1,\cdots,k_{c-2}j$ are clean. This gives a sufficient condition for $d_L$ to be an exact estimator of $s_{ij}^*$.

To estimate the corruption levels of each edge $ij$, CEMP initializes the edge weight of each $ij\in E$ as $w_{ij}^{(0)}=1$. It then iteratively updates the corruption level estimate as the following convex combination of $d_L$'s:

$$s_{ij}^{(t)} = \sum_{L\in N_{ij}^c} w_L^{(t)} d_L / z_{ij}^{(t)} \tag{4}$$

where $z_{ij}^{(t)} = \sum_{L\in N_{ij}^c} w_L^{(t)}$. The cycle weights $w_L^{(t+1)}$ are computed from the edge weights $w_e^{(t+1)} = e^{-\beta_t s_e^{(t)}}$:

$$w_L^{(t+1)} = \prod_{e\in L\setminus\{ij\}} w_e^{(t+1)} = \prod_{e\in L\setminus\{ij\}} e^{-\beta_t s_e^{(t)}}, \tag{5}$$

so that $w_L^{(t+1)}$ focuses on good cycles. The cycle weights and edge corruption levels are alternatingly updated and improved from each other. Interestingly, it is proved in [25] under two different corruption models that CEMP converges linearly to the ground truth corruption estimates under mild conditions for $c=3$. In practice, CEMP only uses 3-cycles for consideration of efficiency. For longer cycles, the complexity of CEMP scales exponentially with the cycle length $c$ (which is discussed in §3.2), and the convergence guarantee of CEMP remains unknown.

## 3. Our method: LongSync

### 3.1. LongSync: Modification of CEMP

Our goal is to develop a scalable variant of CEMP for any fixed number of cycles, $c\geq 3$. The main computational bottleneck of step (4) in CEMP is that computing and summing the cycle inconsistency measures takes $\sum_{ij\in E}|N_{ij}^c| = O(n^c)$ operations and memory. Therefore, to develop a scalable algorithm, we aim to take weighted average over $d_L$ without explicitly computing and storing each $d_L$. To achieve this, we propose the following specification and modification on CEMP:

- **Use Chordal distance on SO(d).** We suggest the distance function

  $$\mathcal{D}(\boldsymbol{R}_1,\boldsymbol{R}_2) = \sqrt{1-\langle \boldsymbol{R}_1,\boldsymbol{R}_2\rangle/d}.$$

  This distance is proportional to the Chordal distance on $SO(3)$, which is the Euclidean distance between two rotations embedded in $\mathbb{R}^{d\times d}$.



Authorized licensed use limited to: University Of Minnesota Duluth. Downloaded on July 29,2025 at 06:11:35 UTC from IEEE Xplore. Restrictions apply.

- **Use weighted quadratic average for corruption level update.** Instead of updating the corruption level estimates by a weighted average of $d_L$, we use the weighted quadratic average of $d_L$, namely

$$s_{ij}^{(t)} = \sqrt{\sum_{L\in N_{ij}^c} w_L^{(t)} d_L^2 / z_{ij}^{(t)}} \tag{6}$$

where the update rule of cycle weights remains the same:

$$w_L^{(t+1)} = \prod_{e\in L\setminus\{ij\}} w_e^{(t+1)} = \prod_{e\in L\setminus\{ij\}} e^{-\beta_t s_e^{(t)}}. \tag{7}$$

As a result, $d_L^2 = d^2(\boldsymbol{R}_L, \boldsymbol{R}_{ij}) = 1 - \langle \boldsymbol{R}_L, \boldsymbol{R}_{ij}\rangle / d$ is linear in both $\boldsymbol{R}_L$ and $\boldsymbol{R}_{ij}$. Therefore one can switch the order of $d^2$ and the weighted summation, so that the computation of $s_{ij}^{(t)}$ can be vectorized. Indeed, by this linearity and equations (4) and (2), and note that $z_{ij}^{(t)} = \sum_{L\in N_{ij}^c} w_L^{(t)}$, we obtain the following equation:

$$\begin{aligned} s_{ij}^{(t)} &= \Big(\sum_{L\in N_{ij}^c} w_L^{(t)} d_L^2 / z_{ij}^{(t)}\Big)^{1/2} \\ &= \Big(\sum_{L\in N_{ij}^c} w_L^{(t)} \mathcal{D}^2(\boldsymbol{R}_L, \boldsymbol{R}_{ij}) / z_{ij}^{(t)}\Big)^{1/2} \\ &= \Big(\mathcal{D}^2\Big(\sum_{L\in N_{ij}^c} w_L^{(t)} \boldsymbol{R}_L, \boldsymbol{R}_{ij}\Big) / z_{ij}^{(t)}\Big)^{1/2} \\ &= \Big(1 - \Big\langle \sum_{L\in N_{ij}^c} w_L^{(t)} \boldsymbol{R}_L, \boldsymbol{R}_{ij}\Big\rangle / d \sum_{L\in N_{ij}^c} w_L^{(t)}\Big)^{1/2} \end{aligned} \tag{8}$$

Equation (8) can be vectorized using the trick of matrix power if we allow repeated nodes for each cycle. That is, one can stack the $s_{ij}^{(t)}$'s and $w_{ij}^{(t)}$'s into matrices $\boldsymbol{S}^{(t)}$ and $\boldsymbol{W}^{(t)}$, and vectorize (8) as

$$\boldsymbol{S}^{(t)} = \Big(\boldsymbol{A} - \Big\langle \Big(\boldsymbol{W}^{(t)} \otimes 1_d \odot \boldsymbol{R}\Big)^{c-1}, \boldsymbol{R}\Big\rangle_{block} \oslash d(\boldsymbol{W}^{(t)})^{c-1}\Big)^{\odot 1/2}. \tag{9}$$

Indeed, by (7) and the definition of $\boldsymbol{R}_L$, $\sum_{L\in C_{ij}^c} w_L^{(t)} \boldsymbol{R}_L$ is the $ij$-th block of $(\boldsymbol{W}^{(t)} \otimes 1_d \odot \boldsymbol{R})^{c-1}$, and $\sum_{L\in C_{ij}^c} w_L^{(t)}$ is the $ij$-th element of $\boldsymbol{W}^{(t)c-1}$, where $C_{ij}^c$ is the set of $c$-cycles containing $ij$ with possibly repeated nodes.

In the case of utilizing only simple cycles, (8) and (9) are not equivalent and we need to correct (9) to remove the cycles with repeated nodes, so that only simple cycles in $N_{ij}^c$ remain. Let $g_c(\boldsymbol{W}, \boldsymbol{R})$ be the matrix valued function where $g_c(\boldsymbol{W}, \boldsymbol{R})(i,j) = \sum_{L\in N_{ij}^c} (\prod_{e\in L\setminus\{ij\}} w_e^{(t)}) \boldsymbol{R}_L$. Let $f_c(\boldsymbol{W})$ be the matrix valued function where $f_c(\boldsymbol{W})(i,j) = \sum_{L\in N_{ij}^c} (\prod_{e\in L\setminus\{ij\}} w_e^{(t)})$. The following result holds:

**Proposition 3.1.** *The update rule of LongSync (8) is equivalent to the following matrix equations:*

$$\boldsymbol{S}^{(t)} = \Big(\boldsymbol{A} - \Big\langle h_c(\boldsymbol{W}^{(t)}, \boldsymbol{R}), \boldsymbol{R}\Big\rangle_{block} \odot \boldsymbol{A}\Big)^{\odot 1/2}, \tag{10}$$

*where* $\boldsymbol{W}^{(t+1)} = \boldsymbol{A} \odot \exp(-\beta_t \boldsymbol{S}^{(t)})$ *and*

$$h_c(\boldsymbol{W}^{(t)}, \boldsymbol{R}) := g_c(\boldsymbol{W}^{(t)}, \boldsymbol{R}) \oslash (d \cdot f_c(\boldsymbol{W}^{(t)}) \otimes 1_d).$$

*Here* exp *denotes the elementwise exponential function.*

We use equation (10) as the update rule of LongSync and propose the vectorized LongSync algorithm in algorithm 1.

**Algorithm 1** (LongSync)

**Input:** pairwise rotation matrix $\boldsymbol{R} \in \mathbb{R}^{dn\times dn}$, adjacency matrix $\boldsymbol{A} \in [0,1]^{n\times n}$, cycle length $c$, positive parameters $\{\beta_t\}_{t\geq 1}$, time step $T$

$\boldsymbol{W}^{(0)}(i,j) \leftarrow \boldsymbol{A}$

**for** $t = 0 : T$ **do**

$$\boldsymbol{S}^{(t)} \leftarrow \Big(\boldsymbol{A} - \Big\langle h_c(\boldsymbol{W}^{(t)}, \boldsymbol{R}), \boldsymbol{R}\Big\rangle_{\text{block}} \odot \boldsymbol{A}\Big)^{\odot 1/2} \tag{11}$$

$$\boldsymbol{W}^{(t+1)} \leftarrow \boldsymbol{A} \odot \exp(-\beta_t \boldsymbol{S}^{(t)}) \tag{12}$$

**end for**

**Output:** edge weights $\boldsymbol{W}^{(T+1)}$, corruption levels $\boldsymbol{S}^{(T)}$

We claim that $g_c$ and $f_c$ can be computed with a sequence of matrix operations, thus greatly reducing the time and space consumption of LongSync compared to its original form. For $c \leq 6$, the time complexity of computing $g_c$ and $f_c$ is $O(r(dn))$, where $r(n)$ is the complexity for multiplying two $n \times n$ matrices; for $c \geq 7$ the time complexity is at most $O(n^{\lfloor (c+3)/2 \rfloor})$. This claim is proved in the supplementary material. We list the formula for $g_c$ and $f_c$ for $c = 3,4,5,6$ inspired by [35, 46]. The formula for $c = 6$ is moved to the supplementary material due to the space limit. For $c \geq 7$ the formula becomes extremely complicated. We remark that in practice, the cycles of length greater than 6 are often not used.

We finally remark that although Algorithm 1 only utilizes cycles of a fixed length, one can easily generalize it to incorporate cycles of different lengths. Indeed, the equation (10) could use a convex combination of $h_c$'s that corresponds to different values of $c$. That is, for a preselected set of cycle lengths $C$, the equation (11) in Algorithm 1 is replaced by

$$\boldsymbol{S}^{(t)} = \Bigg(\boldsymbol{A} - \Bigg\langle \sum_{c\in C} \lambda_c h_c(\boldsymbol{W}^{(t)}, \boldsymbol{R}), \boldsymbol{R}\Bigg\rangle_{\text{block}} \odot \boldsymbol{A}\Bigg)^{\odot 1/2} \tag{13}$$

where the coefficients $\lambda_c$ satisfies $\sum_{c\in C} \lambda_c = 1$ to ensure a convex combination. Here each $\lambda_c$ is user-specified to reflect the importance of the cycles of length $c$. However, the optimal



Authorized licensed use limited to: University Of Minnesota Duluth. Downloaded on July 29,2025 at 06:11:35 UTC from IEEE Xplore. Restrictions apply.

choice of these parameters under certain statistical model remains unclear.

For simplicity, in the experiments we only use a fixed cycle length to avoid choosing $\lambda_c$. We have observed that such simple choice still yields satisfying accuracy in camera orientation estimation on both synthetic and real data. We refer the readers to §5 and 6 for more details.

| $c$ | Formula of $f_c(\boldsymbol{W})$ | Formula of $g_c(\boldsymbol{W},\boldsymbol{R})$ |
|---|---|---|
| 3 | $\boldsymbol{W}^2$ | $\boldsymbol{P}^2$ |
| 4 | $\boldsymbol{W}^3-\mathrm{d}(\boldsymbol{W}^2)\boldsymbol{W}$ <br> $-\boldsymbol{W}\mathrm{d}(\boldsymbol{W}^2)+\boldsymbol{W}^{\odot 3}$ | $\boldsymbol{P}^3-\mathrm{d}_{\text{block}}(\boldsymbol{P}^2)\boldsymbol{P}$ <br> $-\boldsymbol{P}\mathrm{d}_{\text{block}}(\boldsymbol{P}^2)+\boldsymbol{P}^{\odot 3}$ |
| 5 | $\boldsymbol{W}^4-\mathrm{d}(\boldsymbol{W}^3)\boldsymbol{W}$ <br> $-\mathrm{d}(\boldsymbol{W}^2)\boldsymbol{W}^2$ <br> $-\boldsymbol{W}^2\mathrm{d}(\boldsymbol{W}^2)$ <br> $-\boldsymbol{W}\mathrm{d}(\boldsymbol{W}^2)\boldsymbol{W}$ <br> $+3\boldsymbol{W}^{\odot 2}\boldsymbol{W}^2$ <br> $+\boldsymbol{W}\boldsymbol{W}^{\odot 3}+\boldsymbol{W}^{\odot 3}\boldsymbol{W}$ | $\boldsymbol{P}^4-\mathrm{d}_{\text{block}}(\boldsymbol{P}^3)\boldsymbol{P}$ <br> $-\mathrm{d}_{\text{block}}(\boldsymbol{P}^2)\boldsymbol{P}^2$ <br> $-\boldsymbol{P}^2\mathrm{d}_{\text{block}}(\boldsymbol{P}^2)$ <br> $-\boldsymbol{P}\mathrm{d}_{\text{block}}(\boldsymbol{P}^2)\boldsymbol{P}$ <br> $+3\boldsymbol{P}^{\odot 2}\boldsymbol{P}^2$ <br> $+\boldsymbol{P}\boldsymbol{P}^{\odot 3}+\boldsymbol{P}^{\odot 3}\boldsymbol{P}$ |

Table 1. Formulas for $f_c$ and $g_c$. Here we let $\boldsymbol{P}=(\boldsymbol{W}\otimes 1_d)\odot\boldsymbol{R}$ for shorter notation. $\mathrm{d}(\boldsymbol{X})$ returns the diagonal of matrix $\boldsymbol{X}$, $\mathrm{d}_{\text{block}}(\boldsymbol{X})$ returns the diagonal block matrix from the $d\times d$ diagonal blocks of matrix $\boldsymbol{X}$.

### 3.2. Computational Complexity

We derive the space and time complexity for LongSync, and demonstrate its advantages over CEMP. The initialization step involves setting the weights of all edges to 1, which takes time $O(|E|)$ and space $O(n^2)$. For each iteration, LongSync updates the matrices $\boldsymbol{S}^{(t)}$ and $\boldsymbol{W}^{(t)}$ with equations (11) and (12), respectively. Computing $\boldsymbol{W}^{(t+1)}$ involves two matrix subtractions, one scalar-matrix multiplication and one element-wise matrix exponential operation on $\boldsymbol{S}^{(t)}\in\mathbb{R}^{n\times n}$. Therefore the update step (12) takes at most $O(n^2)$ time and space. Equation (11), on the other hand, involves a sequence of matrix operations on $\boldsymbol{P}^{(t)}=(\boldsymbol{W}^{(t)}\otimes 1_d)\odot\boldsymbol{R}\in\mathbb{R}^{dn\times dn}$ and $\boldsymbol{W}^{(t)}\in\mathbb{R}^{n\times n}$, including matrix multiplications, element-wise multiplications and diagonal block selections. Computing $\boldsymbol{P}^{(t)}$ takes $O(d^2n^2)$ memory and $O(d^2n^2)$ time. The matrix operations on $\boldsymbol{P}^{(t)}$ take $O(K_cd^3n^3)$ time and $O(d^2n^2)$ space, and the matrix operations on $\boldsymbol{W}^{(t)}$ take $O(K_cn^3)$ time and $O(n^2)$ space, where $K_c$ is the number of terms in the equation for $f_c$ and $g_c$. Therefore, each iteration over $t$ takes $O(K_cd^3n^3)$ time and $O(d^2n^2)$ space. To sum up, for LongSync, the time complexity is $O(K_cd^3n^3)$ and the space complexity is $O(d^2n^2)$. For $c=3,4,5,6$, the number $K_c$ is equal to 1,3,9,32.

In comparison, we consider the initialization step of CEMP. For each edge $ij\in E$ and $L\in N_{ij}^c$, initializing CEMP involves computing and storing all the cycle inconsistency measures $d_L$ using equation (2). For each $L\in N_{ij}^c$, computing $d_L$ involves multiplying $c$ rotations, which takes $O(cd^3)$ time and $O(d^2)$ space. This step is repeated for each $ij\in E$ and $L\in N_{ij}^c$, therefore the total time complexity is $O(cd^3\sum_{ij\in E}|N_{ij}^c|)$ and the total space complexity is $O(d^2\sum_{ij\in E}|N_{ij}^c|)$. Since for each edge there are $(n-2)(n-3)\cdots(n-c+1)=O(n^{c-2})$ cycle candidates, we know that $|N_{ij}^c|\sim O(n^{c-2})$ for each $ij\in E$ in the worst case scenario of a dense graph. Therefore the initialization of CEMP takes $O(cd^3n^{c-2}|E|)$ time and $O(d^2n^{c-2}|E|)$ space in the worst case. Given $c\geq 4$ and $|E|\sim n^2$, CEMP requires much more time and space than LongSync.

## 4. Theory for Uniform Corruption Model

In this section, we present the exact recovery guarantee of LongSync under the uniform corruption model (UCM). UCM is a popular probabilistic model that is widely adopted for synthetic experiments of many previous works on group synchronization [8, 25, 28, 29, 32, 42]. The model UCM($n$,$p$,$q_g$) assumes that $G$ is an Erdős-Rényi graph with edge connection probability $p$. For each edge $ij\in E$, $\boldsymbol{R}_{ij}$ is generated independently as follows:

$$\boldsymbol{R}_{ij}=\begin{cases}\boldsymbol{R}_{ij}^* & \text{w.p. } q_g;\\ \tilde{\boldsymbol{R}}_{ij}\sim\text{Haar}(\mathcal{G}) & \text{w.p. } 1-q_g.\end{cases}$$

We also developed an exact recovery theory for a general model of adversarial corruption, which we include in section C.1 of the supplementary material. An informal version of our main result for UCM is stated in Theorem 4.1. Although the application of this paper is focused on rotation synchronization, the following theory for UCM is valid for any compact group $\mathcal{G}$, as explained in the supplementary material.

**Theorem 4.1.** *Let $0<r<1$, $0<q<1$, $0<p\leq 1$, $\mathcal{G}=SO(3)$. Assume LongSync is applied with cycles of length $c$, $n/\log n\sim p^{-(c-1)/(c-2-\epsilon)}q_g^{-7(c-1)/3(c-2)}$ for some $\epsilon>0$ and*

$$1/\beta_{t+1}=r/\beta_t \textit{ for all } t\geq 1.$$

*Then with appropriate choices of $\beta_0$,$\beta_1$,$r$, and high probability, $\max_{ij\in E}|s_{ij}^*-s_{ij}^{(t)}|\leq\frac{1}{2c\beta_t}$ for all $t\geq 1$.*

The major difficulty of proving Theorem 4.1 is the dependence in the cycle inconsistency measures for cycles in $N_{ij}^c$ when $c\geq 4$. Unlike the 3-cycle case, the cycle inconsistency measure of a 4-cycle $L_1=(ik_1,k_1k_2,k_2j)$ is correlated with that of $L_2=(ik_1,k_1k_3,k_3j)$ under UCM. Therefore the key concentration inequalities for the proof cannot be concluded from the standard Chernoff bounds. To overcome this theoretical obstacle, we have integrated various mathematical techniques from [3, 7, 22–24, 48] to prove the theorem, whose details are included in the supplementary material.

Theorem 4.1 provides an upper bound of the sample complexity (the required graph size $n$) of LongSync for exact recovery of the ground truth solutions. This sample complexity is the closest to the information theoretic bound among all existing rotation synchronization methods. The comparison with previous works is summarized in Table 2.



Authorized licensed use limited to: University Of Minnesota Duluth. Downloaded on July 29,2025 at 06:11:35 UTC from IEEE Xplore. Restrictions apply.

| Reference | Sample Complexity |
|---|---|
| [25] for CEMP | $O(p^{-2}q_g^{-28/3})$ |
| [29] for ReSync | $O(p^{-2}q_g^{-7})$ |
| Ours for CEMP | $O(p^{-2-\epsilon}q_g^{-14/3})$ |
| Ours for LongSync ($c$=3) | $O(p^{-2-\epsilon}q_g^{-14/3})$ |
| Ours for LongSync ($c$=4) | $O(p^{-1.5-\epsilon}q_g^{-3.5})$ |
| Ours for LongSync (any $c$) | $O\left(p^{-\frac{c-1}{c-2}-\epsilon}q_g^{-\frac{7(c-1)}{3(c-2)}}\right)$ |
| Ours for LongSync ($c\to\infty$) | $O(p^{-1-\epsilon}q_g^{-7/3})$ |
| Information Theoretic Bound [6] | $O(p^{-1}q_g^{-2})$ |

Table 2. Comparison of the sample complexity requirement. Lower absolute values of the powers on $p,q_g$ indicate better results. $\epsilon$ is an arbitrarily small positive real number.

## 5. Synthetic Data Experiment

We test LongSync on synthetic datasets generated with Uniform Corruption Model (UCM) and Uniform Bipartite Corruption Model (UBCM) respectively described in §5.1 and 5.2. For both models with their corresponding viewing graphs $G=([n],E)$, we sample the ground truth absolute rotation matrices $\{\boldsymbol{R}_i^*\}_{i\in[n]}$ independently from the Haar measure on SO(3), and we generate the observed relative rotations $\{\boldsymbol{R}_{ij}\}_{ij\in E}$ independently as follows:

$$\boldsymbol{R}_{ij}=\begin{cases}\boldsymbol{R}_i^*\boldsymbol{R}_j^* & \text{w.p. } q_g;\\ \tilde{\boldsymbol{R}}_{ij}\sim\text{Haar}(SO(3)) & \text{w.p. } 1-q_g.\end{cases}$$

We use LongSync with cycle length $c$, $\beta_t=\min(2^t,20)$ and $T=10$ and record the edge weights. For UCM we set $c=4,5$ and for UBCM we only use $c=4$ since no 5-cycles exist. For our method, we first build a weighted graph whose edge weights are estimated by LongSync. We then extract a maximum spanning tree (MST) of the resulting weighted graph. The resulting spanning tree is expected to be the cleanest possible spanning tree. To initialize our solution of absolute rotations, we first fix $\boldsymbol{R}_1$ as the identity rotation, and find the rest of $\boldsymbol{R}_i$'s by consecutively multiplying the relative rotations along the spanning tree using the formula $\boldsymbol{R}_i=\boldsymbol{R}_{ij}\boldsymbol{R}_j$. To refine our initialized solution, we apply IRLS with Geman-McClure [4] loss functions to minimize $\sum_{ij\in E}\rho_{GM}(d_\angle(\boldsymbol{R}_{ij},\boldsymbol{R}_i\boldsymbol{R}_j^T))$, where $d_\angle$ is the geodesic distance in $SO(3)$. We refer to this method as LongSync+IRLS.

To demonstrate the advantages of utilizing longer cycle information, we compare our method with IRLS initialized by other two different spanning trees. The first one is the random spanning tree, which uses no cycle information. The other one the MST extracted from the CEMP-estimated weights. Note that CEMP only uses 3-cycle information. We refer to these methods as IRLS and CEMP+IRLS respectively.

Since the solution of absolute rotations is determined up to a global rotation, we align our estimated rotation $\{\hat{\boldsymbol{R}}_i\}$ with $\{\boldsymbol{R}_i^*\}$ by $\boldsymbol{R}_{\text{align}}$ that minimizes the $\ell_1$ rotation alignment error $\sum_{i\in[n]}\|\hat{\boldsymbol{R}}_i\boldsymbol{R}_{\text{align}}-\boldsymbol{R}_i^*\|_F$. We report the mean estimation error in degrees: $180\cdot\sum_{i\in[n]}d_\angle(\hat{\boldsymbol{R}}_i\boldsymbol{R}_{\text{align}},\boldsymbol{R}_i^*)/n$.

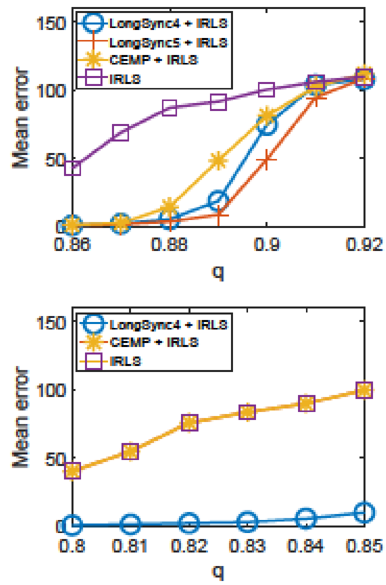


Figure 1. Average errors for IRLS, CEMP+IRLS and LongSync +IRLS with $c=4$, 5, using the uniform corruption (top) and uniform bipartite corruption (bottom) models. The mean errors are measured in degrees. LongSync4 and LongSync5 refer to LongSync with 4 and 5 cycles, respectively.

### 5.1. Uniform Corruption Model

For UCM($n$,$p$,$q_g$), we take $n=200$ and $p=1$ and corruption probability $q=1-q_g$ ranges from 0.86 to 0.92. We report the average mean estimation error from 20 trials of the uniform corruption model in the top panel of Figure 1.

We note that LongSync uniformly improves IRLS, and the mean error of LongSync decreases as the cycle length increases. When $q>0.86$, the expected number of clean 3-cycles for each edge is less than 4, and therefore longer cycles are helpful. The numerical result aligns with our theory that using longer cycles may tolerate higher corruption with fixed graph size $n$.

### 5.2. Uniform Bipartite Corruption Model

For UBCM, we first generate the graph and relative rotations by UCM($n$,$p$,$q_g$) with $n=200$, $p=1$ and $q=1-q_g$ ranging from 0.8 to 0.85. Then we split the nodes into two clusters of equal size and remove the intra-cluster edges for both clusters. The resulting graph is bipartite, where only cycles of even lengths exist. We report the mean estimation error from 20 trials in the bottom panel of Figure 1.

We observe that LongSync with 4-cycles almost exactly recovers the rotations, while for other algorithms the rotation estimates are not even close to the ground truth.



Authorized licensed use limited to: University Of Minnesota Duluth. Downloaded on July 29,2025 at 06:11:35 UTC from IEEE Xplore. Restrictions apply.

## 6. Real Data Experiment

We test distributed synchronization with LongSync on the PhotoTourism dataset [50] to demonstrate its advantages in accuracy and speed over other baselines. PhotoTourism is a large scale dataset consisting of 15 sets of images taken for 3D reconstruction. The smallest dataset consists of 247 cameras, and the largest dataset consists of 5433 cameras. The input graph and initial pairwise rotation estimates are provided in the dataset. In the following, we first explain the common steps for distributed synchronization, and our improvement using LongSync. We then describe our graph processing method for filtering bad nodes and edges, which also is applicable to other baseline methods.

**Steps in Distributed Synchronization:**

1. **Graph partitioning.** The first step involves partitioning the graph $G=([n],E)$ into $K$ clusters $G_i=(V_i,E_i), i\in[K]$. In this paper we apply spectral clustering algorithm [37] on the adjacency matrix $G$, where $K = 0.6\sqrt{np}$ and $p=2|E|/(n(n-1))$.
2. **Synchronization within clusters.** Run standard synchronization solvers for each cluster. In this work, we use the current state-of-the-art method MPLS [39]. Note that for each camera $p$ in cluster $k$, one can only estimate the true rotation $\boldsymbol{R}_p^*$ up to a global rotation $\boldsymbol{R}_k$. Namely, one only obtains $\hat{\boldsymbol{R}}^p \approx \boldsymbol{R}_p^* \boldsymbol{R}_k^{-1}$ where $\boldsymbol{R}_k$ is unknown and is the same for all cameras in cluster $k$.
3. **Estimation of inter-cluster rotations.** To find $\boldsymbol{R}_p^*$ of all cameras, one needs to solve $\boldsymbol{R}_k$ for all clusters. Namely, one needs to rotate and stitch the solutions of all clusters so that they are in the same reference frame. To do this, it is common to first estimate the inter-clsuter rotations $\boldsymbol{R}_{kl} := \boldsymbol{R}_k \boldsymbol{R}_l^{-1}$ between pairs of clusters $k,l$, and then synchronize these relative rotations. To estimate each $\boldsymbol{R}_{kl}$, we note that $\boldsymbol{R}_{kl} = \boldsymbol{R}_p^{*-1} \boldsymbol{R}_{pq}^* \boldsymbol{R}_q^*$ for each $p\in V_k$ and $q\in V_l$. Therefore, one can use the rotations in the set $S_{kl} := \{\hat{\boldsymbol{R}}_p^{-1} \boldsymbol{R}_{pq} \hat{\boldsymbol{R}}_q\}_{p\in V_k, q\in V_l}$ to approximate $\boldsymbol{R}_{kl}$. We remark that this step is crucial to the overall performance of distributed methods, and we compare the following methods for solving $\boldsymbol{R}_{kl}$:
   - MultSync [9]: Run synchronization on a multi-graph where each edge $kl$ is assigned a set of relative rotations $\{\hat{\boldsymbol{R}}_p^{-1} \boldsymbol{R}_{pq} \hat{\boldsymbol{R}}_q\}_{p\in V_k, q\in V_l}$. This combines the step 3 and 4 in a unified least squares formulation.
   - Edge averaging using IRLS: We initialize $\hat{\boldsymbol{R}}_{kl}$ with the quaternion $\ell_2$ mean of the set $S_{kl}$ and refine it using $\ell_1$-rotation averaging [19]. We refer to this method as IRLS in our comparison.
   - Edge averaging using LongSync: We first perform LongSync with 4-cycles to estimate the weights of these inter-cluster edges (there are no 3-cycles for a bipartite graph). We next initialize $\hat{\boldsymbol{R}}_{kl}$ as the quaternion weighted $\ell_2$ mean of $S_{kl}$, using the edge weights from LongSync by their LongSync weights. Lastly, we refine the solution using [19].
4. **Synchronization of inter-cluster rotations.** This step is skipped for MultiSync. For other methods described in step 3, we find $\boldsymbol{R}_k$ (up to a rotation) for each cluster $k$ from the estimated $\{\boldsymbol{R}_{kl}\}_{k,l\in[K]}$ by MPLS.
5. **Rotation merging.** Finally, for each camera $p$ in cluster $k$, the rotation estimate of $p$ is given by $\boldsymbol{R}_p^{\text{final}} = \hat{\boldsymbol{R}}_p \boldsymbol{R}_k^{-1}$.

Next, we introduce our graph processing method to further boost the performance of all tested methods.

**Extra Improvement by Graph Processing:**

- **Spectral clustering with Jaccard Index.** For step 1, we use the Jaccard index matrix as the similarity matrix for spectral clustering, instead of the adjacency matrix. The $n\times n$ Jaccard index matrix $\boldsymbol{A}_J$ is defined as follows:

$$\boldsymbol{A}_J(i,j) = \begin{cases} 0 & ij \notin E \\ \frac{|N_i \cap N_j|}{|N_i \cup N_j|} & ij \in E \end{cases} \tag{14}$$

  where $N_i$ and $N_j$ denote the sets of neighboring nodes of node $i$ and $j$, respectively. In this way, $\boldsymbol{A}_J(i,j)$ is higher for the pair $ij$ contained in many 3-cycles, which is a more robust and nicely scaled statistics ($\in[0,1]$) for measuring the local graph density around edge $ij$.
- **Refinement of intra-cluster edges and nodes.** For step 2, after the MPLS step, we perform CEMP with 3-cycles to estimate the corruption level of the intra-cluster edges for each cluster. We remove a camera if the number of neighboring 'good' edges, i.e. the edges with corruption level less than 0.1, is less than 4. The numbers 4 and 0.1 are chosen to balance the number of remaining cameras and the quality of intra-cluster rotation estimates. In order to eliminate the sparsely connected components inside the cluster, we use the Matlab built-in hierarchical spectral clustering function on the remaining cameras with the 'cutoff' and 'depth' parameters as 2 and 4, and we keep the largest component. The absolute rotations for the remaining cameras are estimated by MPLS. We remark that one could replace CEMP by LongSync with 3-cycles. However, we have not observed significant difference in the performance.

We respectively name MultiSync and IRLS with our new graph processing method as MultiSync(New) and IRLS(new). "LongSync" in our experiment refers to the full version of our algorithm: use LongSync weights for edge averaging in Step 3, with the graph processing step. We also compare with MPLS on the whole dataset, since it is a state-of-the-art non-distributed method, but we note that MPLS is significantly slower than all distributed methods. We report median error $180 \cdot \text{median}(\{d_\angle(\hat{\boldsymbol{R}}_i \boldsymbol{R}_{\text{align}}, \boldsymbol{R}_i^*)\}_{i\in[n]}$ of the tested methods on 14 datasets in Figure 2. We exclude the result of Gendarmenmarkt since all methods return large estimation errors in the figure. The full results, including that of mean error are included in the supplementary material.

In Figure 3, for each distributed method, we report the ratio (in percentage) between its total runtime on all datasets and that of the non-distributed MPLS. Namely, we compute



Authorized licensed use limited to: University Of Minnesota Duluth. Downloaded on July 29,2025 at 06:11:35 UTC from IEEE Xplore. Restrictions apply.

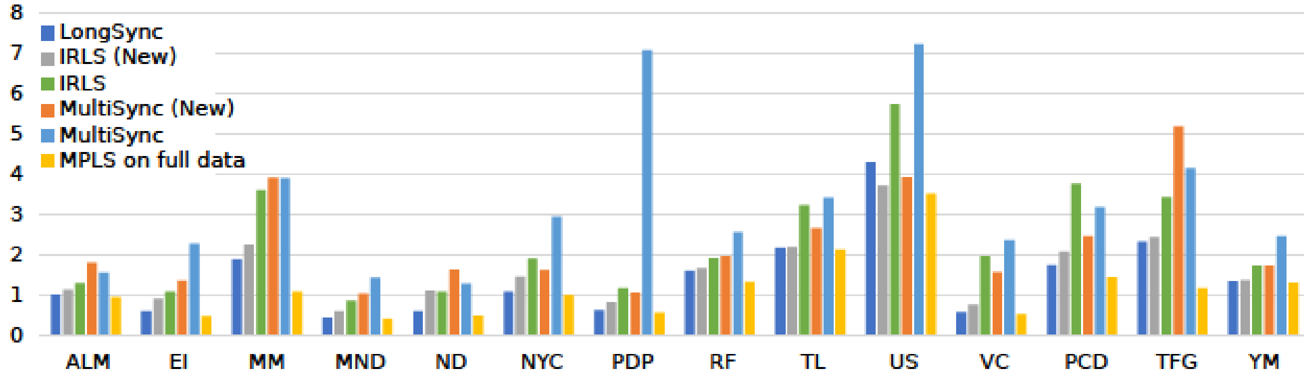


Figure 2. Median error for rotations for each dataset measured in degrees.

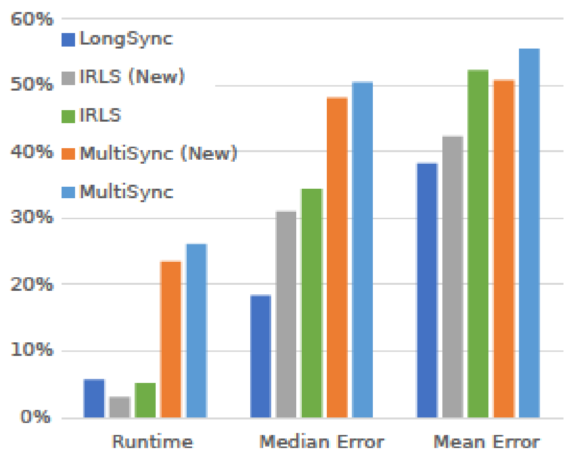


Figure 3. Runtime ratio and average median and mean error gaps between the distributed methods and MPLS on the entire graph.

$\sum_{d\in D} t_{\text{dist},d} / \sum_{d\in D} t_{\text{MPLS},d}$, where $D$ is the set of 15 datasets, and each $t_{\text{dist},d}$ and $t_{\text{MPLS},d}$ is respectively the runtime of the distributed method and MPLS on data $d$. In the same figure, we present the mean/median error gap between each distributed method and MPLS. The mean and median error gap is respectively defined as $(\bar{e}_{\text{dist}} - \bar{e}_{\text{MPLS}})/\bar{e}_{\text{dist}}$ and $(\hat{e}_{\text{dist}} - \hat{e}_{\text{MPLS}})/\hat{e}_{\text{dist}}$, where $\bar{e}$ and $\hat{e}$ respectively denote the mean and median error over all cameras.

From Figure 2 and 3, our method outperforms other distributed methods on 13 out of 15 datasets. The most significant improvement in mean error and median error are respectively 28.6% and 46.4% (in Notre Dame) compared to the best performing method between IRLS(new) and MultiSync(new). The improvement is even more significant when comparing to the original version of these baseline methods without our graph processing method. The only two datasets without improvement are Gendarmenmarkt and Union Square. Our method is comparable to others on Union Square, and all methods return large errors on Gendarmenmarkt due to many repetitive patterns in its 3D scene.

The average mean and median error gap between our method and full MPLS are respectively 38.3% and 18.4%. Compared to the best performing method among others, our method reduces the average median error gap by 40.8%, and the average mean error gap by 9.6%. In terms of runtime, our method is uniformly faster than MultiSync and it is scalable on the largest dataset, taking less than 6% of the total runtime of full MPLS. In conclusion, our method significantly improves the result of distributed rotation synchronization without compromising runtime.

In the supplementary material, we further demonstrate the improvement by our new graph processing method, which significantly improves the results of LongSync (without extra graph processing) in 14 of the 15 datasets. On these 14 datasets, the average reduction on mean error is 59.2% and the average reduction on median error is 28.5%.

## 7. Conclusion

We propose LongSync, a robust and efficient algorithm for group synchronization. It modifies and vectorizes CEMP which enables efficient computation when using longer cycles. The theory we developed for LongSync is the strongest among all other existing results under UCM. Experiment shows that LongSync, together with our improved graph preprocessing method, achieves superior accuracy for distributed synchronization on large real datasets with competitive runtime. However, our method also has some limitations. First of all, in theory there is still a small gap of sample complexity from our method to the information theoretic one. Filling this gap is an open problem, which requires new tools and possibly more sophisticated analysis. Second, our graph preprocessing method is quite heuristic, and an automatic way of choosing parameters is needed. Our work also opens a door for some important future directions, including distributed partial permutation synchronization for multi-image matching, angular synchronization for Cryo-EM and Jigsaw Puzzles, and analysis of their algorithms.



Authorized licensed use limited to: University Of Minnesota Duluth. Downloaded on July 29,2025 at 06:11:35 UTC from IEEE Xplore. Restrictions apply.

## References


[1] Federica Arrigoni, Beatrice Rossi, Pasqualina Fragneto, and Andrea Fusiello. Robust synchronization in so(3) and se(3) via low-rank and sparse matrix decomposition. *Computer Vision and Image Understanding*, 174:95–113, 2018. 2

[2] A. S. Bandeira, N. Boumal, and A. Singer. Tightness of the maximum likelihood semidefinite relaxation for angular synchronization. *arXiv preprint arXiv:1411.3272*, 2014. 2

[3] Olivier Bousquet. A Bennett concentration inequality and its application to suprema of empirical processes. *C. R. Math. Acad. Sci. Paris*, 334(6):495–500, 2002. 5, 14, 16, 21

[4] Avishek Chatterjee and Venu Madhav Govindu. Efficient and robust large-scale rotation averaging. In *IEEE International Conference on Computer Vision, ICCV 2013, Sydney, Australia, December 1-8, 2013*, pages 521–528, 2013. 2, 6

[5] Avishek Chatterjee and Venu Madhav Govindu. Robust relative rotation averaging. *IEEE Trans. Pattern Anal. Mach. Intell.*, 40(4):958–972, 2018. 2

[6] Yuxin Chen, Changho Suh, and Andrea J. Goldsmith. Information recovery from pairwise measurements. *IEEE Trans. Inf. Theory*, 62(10):5881–5905, 2016. 6

[7] Keresztély Corradi and András Hajnal. On the maximal number of independent circuits in a graph. *Acta Mathematica Hungarica*, 14(3-4):423–439, 1963. 5, 14

[8] Mihai Cucuringu. Synchronization over Z2 and community detection in signed multiplex networks with constraints. *J. Complex Networks*, 3(3):469–506, 2015. 5

[9] Andrea Porfiri Dal Cin, Luca Magri, Federica Arrigoni, Andrea Fusiello, and Giacomo Boracchi. Synchronization of group-labelled multi-graphs. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 6433–6443, 2021. 2, 7

[10] Frank Dellaert, David M. Rosen, Jing Wu, Robert E. Mahony, and Luca Carlone. Shonan rotation averaging: Global optimality by surfing so(p)$^n$. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part VI*, volume 12351 of *Lecture Notes in Computer Science*, pages 292–308. Springer, 2020. 2

[11] Anders Eriksson, Carl Olsson, Fredrik Kahl, and Tat-Jun Chin. Rotation averaging and strong duality, 2017.

[12] Anders P. Eriksson, Carl Olsson, Fredrik Kahl, and Tat-Jun Chin. Rotation averaging and strong duality. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pages 127–135. Computer Vision Foundation / IEEE Computer Society, 2018.

[13] Anders P. Eriksson, Carl Olsson, Fredrik Kahl, and Tat-Jun Chin. Rotation averaging with the chordal distance: Global minimizers and strong duality. *IEEE Trans. Pattern Anal. Mach. Intell.*, 43(1):256–268, 2021. 2

[14] Xiang Gao, Hainan Cui, Menghan Li, Zexiao Xie, and Shuhan Shen. Irav3: Hierarchical incremental rotation averaging on the fly. *IEEE Trans. Circuits Syst. Video Technol.*, 33(4):2001–2006, 2023. 2

[15] Xiang Gao, Hainan Cui, and Shuhan Shen. Incremental rotation averaging revisited and more: A new rotation averaging benchmark. *arXiv preprint*, abs/2309.16924, 2023.

[16] Xiang Gao, Lingjie Zhu, Hainan Cui, Zexiao Xie, and Shuhan Shen. IRA++: distributed incremental rotation averaging. *IEEE Trans. Circuits Syst. Video Technol.*, 32(7):4885–4892, 2022. 2

[17] Thomas Goldstein, Paul Hand, Choongbum Lee, Vladislav Voroninski, and Stefano Soatto. Shapefit and shapekick for robust, scalable structure from motion. In *Computer Vision - ECCV 2016 - 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part VII*, pages 289–304, 2016. 1

[18] V.M. Govindu. Lie-algebraic averaging for globally consistent motion estimation. In *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.*, volume 1, pages I–I, 2004. 2

[19] R. Hartley, K. Aftab, and J. Trumpf. L1 rotation averaging using the weiszfeld algorithm. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 3041–3048. IEEE, 2011. 7

[20] Richard I. Hartley, Khurrum Aftab, and Jochen Trumpf. L1 rotation averaging using the weiszfeld algorithm. In *The 24th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2011, Colorado Springs, CO, USA, 20-25 June 2011*, pages 3041–3048, 2011. 2

[21] Xiangru Huang, Zhenxiao Liang, Xiaowei Zhou, Yao Xie, Leonidas J. Guibas, and Qixing Huang. Learning transformation synchronization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 2

[22] Svante Janson. Poisson approximation for large deviations. *Random Structures & Algorithms*, 1(2):221–229, 1990. 5, 14, 16

[23] Svante Janson. Large deviations for sums of partly dependent random variables. *Random Structures & Algorithms*, 24(3):234–248, 2004. 18, 21

[24] Jeong Han Kim and Van H Vu. Concentration of multivariate polynomials and its applications. *Combinatorica*, 20(3):417–434, 2000. 5, 14, 18

[25] Gilad Lerman and Yunpeng Shi. Robust group synchronization via cycle-edge message passing. *Foundations of Computational Mathematics*, 22(6):1665–1741, 2022. 1, 2, 3, 5, 6, 13, 14, 16, 19, 21

[26] Heng Li, Zhaopeng Cui, Shuaicheng Liu, and Ping Tan. RAGO: recurrent graph optimizer for multiple rotation averaging. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*, pages 15766–15775. IEEE, 2022. 2

[27] Shaohan Li, Yunpeng Shi, and Gilad Lerman. Fast, accurate and memory-efficient partial permutation synchronization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 15735–15743, June 2022. 2

[28] Shuyang Ling. Near-optimal performance bounds for orthogonal and permutation group synchronization via spectral methods. *arXiv preprint arXiv:2008.05341*, 2020. 5

[29] Huikang Liu, Xiao Li, and Anthony Man-Cho So. Resync: Riemannian subgradient-based robust rotation synchronization. In *Advances in Neural Information Processing Systems: 37th Annual Conference on Neural Information Processing Systems*, 2023. 2, 5, 6

[30] Tyler Maunu and Gilad Lerman. Depth descent synchronization in SO(D). *Int. J. Comput. Vis.*, 131(4):968–986, 2023. 2

[31] Onur Özyesil and Amit Singer. Robust camera location estimation by convex programming. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2674–2683, 2015. 1

[32] Deepti Pachauri, Risi Kondor, and Vikas Singh. Solving the multi-way matching problem by permutation synchronization. In




Authorized licensed use limited to: University Of Minnesota Duluth. Downloaded on July 29,2025 at 06:11:35 UTC from IEEE Xplore. Restrictions apply.

C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 1860–1868. Curran Associates, Inc., 2013. 5

[33] Álvaro Parra, Shin-Fang Ch'ng, Tat-Jun Chin, Anders P. Eriksson, and Ian Reid. Rotation coordinate descent for fast globally optimal rotation averaging. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*, pages 4298–4307. Computer Vision Foundation / IEEE, 2021. 2

[34] Pulak Purkait, Tat-Jun Chin, and Ian Reid. Neurora: Neural robust rotation averaging, 2020. 2

[35] Ian C Ross and Frank Harary. On the determination of redundancies in sociometric chains. *Psychometrika*, 17(2):195–208, 1952. 4, 12

[36] Tianwei Shen, Siyu Zhu, Tian Fang, Runze Zhang, and Long Quan. Graph-based consistent matching for structure-from-motion. In *European Conference on Computer Vision*, pages 139–155. Springer, 2016. 2

[37] Jianbo Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905, 2000. 7

[38] Yunpeng Shi and Gilad Lerman. Estimation of camera locations in highly corrupted scenarios: All about that base, no shape trouble. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pages 2868–2876, 2018. 2

[39] Yunpeng Shi and Gilad Lerman. Message passing least squares framework and its application to rotation synchronization. In *Proceedings of the 37th International Conference on Machine Learning (ICML)*, 2020. 1, 2, 7

[40] Yunpeng Shi, Shaohan Li, and Gilad Lerman. Robust multi-object matching via iterative reweighting of the graph connection laplacian. *Advances in Neural Information Processing Systems*, 2020-December, 2020. 2

[41] Chitturi Sidhartha and Venu Madhav Govindu. It is all in the weights: Robust rotation averaging revisited. In *International Conference on 3D Vision, 3DV 2021, London, United Kingdom, December 1-3, 2021*, pages 1134–1143. IEEE, 2021. 2

[42] Amit Singer. Angular synchronization by eigenvectors and semidefinite programming. *Applied and computational harmonic analysis*, 30(1):20–36, 2011. 2, 5

[43] A. Singer and Y. Shkolnisky. Angular synchronization by eigenvectors and semidefinite programming: Analysis and application to class averaging in cryo-electron microscopy, 2009. 2

[44] Johan Thunberg, Florian Bernard, and Jorge Goncalves. Distributed methods for synchronization of orthogonal matrices over graphs. *Automatica*, 80:243–252, jun 2017. 2

[45] Roberto Tron and René Vidal. Distributed image-based 3-d localization of camera sensor networks. In *Proceedings of the 48th IEEE Conference on Decision and Control, CDC 2009, December 16-18, 2009, Shanghai, China*, pages 901–908, 2009. 2

[46] Anton Voropaev and Sergey Perepechko. The number of fixed length cycles in undirected graph. explicit formulae in case of small lengths. *Bulletin of PFUR. Series Mathematics, Information Sciences, Physics*, pages 5–11, 01 2012. 4

[47] Anton Voropaev and Sergey Perepechko. The number of fixed length cycles in undirected graph. explicit formulae in case of small lengths. *Bulletin of PFUR. Series Mathematics, Information Sciences, Physics*, pages 5–11, 01 2012. 13

[48] Van H Vu. A large deviation result on the number of small subgraphs of a random graph. *Combinatorics, Probability and Computing*, 10(1):79–94, 2001. 5, 14, 17

[49] Lanhui Wang and Amit Singer. Exact and stable recovery of rotations for robust synchronization. *Information and Inference*, 2013. 2

[50] Kyle Wilson and Noah Snavely. Robust global translations with 1dsfm. In *Computer Vision - ECCV 2014 - 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part III*, pages 61–75, 2014. 7

[51] Christopher Zach, Manfred Klopschitz, and Marc Pollefeys. Disambiguating visual relations using loop constraints. In *The Twenty-Third IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2010, San Francisco, CA, USA, 13-18 June 2010*, pages 1426–1433, 2010. 2



Authorized licensed use limited to: University Of Minnesota Duluth. Downloaded on July 29,2025 at 06:11:35 UTC from IEEE Xplore. Restrictions apply.

# Supplementary Material

## A. Full Results for the Real Data Experiment

We record the full results for our real data experiment in Tables 3 and 4.

| Data | | | LongSync | | | MultiSync-New | | | IRLS-New | | | MPLS on full dataset | | | Remaining |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | $n$ | $k$ | $\bar{e}$ | $\hat{e}$ | $t$ | $\bar{e}$ | $\hat{e}$ | $t$ | $\bar{e}$ | $\hat{e}$ | $t$ | $\bar{e}$ | $\hat{e}$ | $t$ | cameras |
| Alamo | 627 | 10 | **2.67** | **1.03** | 6.68 | 3.21 | 1.81 | 80.77 | 2.74 | 1.14 | 3.58 | 2.03 | 0.95 | 53.67 | 0.70 |
| Ellis Island | 247 | 7 | **1.07** | **0.60** | 1.05 | 1.81 | 1.37 | 24.46 | 1.29 | 0.92 | 0.79 | 0.89 | 0.48 | 4.40 | 0.66 |
| Madrid Metropolis | 394 | 6 | **2.98** | **1.89** | 1.28 | 4.42 | 3.92 | 14.76 | 3.35 | 2.27 | 1.00 | 2.10 | 1.10 | 5.76 | 0.53 |
| Montreal Notre Dame | 474 | 8 | **3.89** | **0.45** | 2.96 | 4.34 | 1.05 | 39.17 | 4.00 | 0.60 | 1.86 | 0.78 | 0.41 | 14.95 | 0.70 |
| Notre Dame | 553 | 11 | **1.00** | **0.60** | 5.26 | 1.93 | 1.64 | 111.98 | 1.40 | 1.12 | 3.04 | 0.96 | 0.50 | 52.68 | 0.66 |
| NYC Library | 376 | 6 | **2.05** | **1.09** | 1.20 | 2.56 | 1.63 | 14.57 | 2.31 | 1.47 | 0.83 | 1.56 | 1.01 | 5.02 | 0.56 |
| Piazza del Popolo | 345 | 7 | **3.83** | **0.62** | 1.39 | 4.07 | 1.07 | 22.91 | 3.89 | 0.83 | 0.98 | 1.61 | 0.57 | 6.26 | 0.61 |
| Roman Forum | 1102 | 6 | **2.47** | **1.61** | 5.60 | 2.84 | 1.98 | 16.13 | 2.55 | 1.68 | 3.21 | 1.80 | 1.33 | 24.05 | 0.48 |
| Tower of London | 489 | 5 | **2.85** | **2.18** | 1.82 | 3.49 | 2.66 | 8.43 | 2.94 | 2.20 | 1.17 | 2.50 | 2.13 | 5.63 | 0.60 |
| Union Square | 930 | 4 | 8.02 | 4.31 | 2.59 | 7.79 | 3.93 | 4.92 | **7.76** | **3.73** | 1.73 | 4.50 | 3.53 | 7.20 | 0.42 |
| Vienna Cathedral | 918 | 9 | **3.65** | **0.58** | 5.66 | 4.34 | 1.57 | 56.45 | 3.76 | 0.77 | 3.36 | 1.30 | 0.53 | 54.62 | 0.48 |
| Gendarmenmarkt | 742 | 6 | 84.95 | **77.60** | 3.21 | 83.30 | 80.48 | 15.75 | **74.71** | 84.23 | 2.27 | 48.52 | 40.16 | 13.58 | 0.47 |
| Piccadilly | 2508 | 9 | **5.07** | **1.75** | 22.56 | 5.43 | 2.46 | 65.89 | 5.21 | 2.08 | 10.72 | 2.20 | 1.45 | 429.40 | 0.45 |
| Trafalgar | 5433 | 9 | **7.12** | **2.33** | 79.33 | 10.01 | 5.19 | 91.69 | 7.25 | 2.44 | 41.59 | 1.88 | 1.17 | 1796.41 | 0.38 |
| Yorkminster | 458 | 6 | **1.97** | **1.36** | 2.52 | 2.33 | 1.74 | 14.97 | 2.01 | 1.38 | 1.89 | 1.63 | 1.31 | 7.05 | 0.61 |

Table 3. Results for PhotoTourism. For each dataset, $\bar{e}$ and $\hat{e}$ indicate the mean error and median error of the output absolute rotation estimates measured in degrees, and $t$ is the total runtime of each method measured in seconds. The last column indicates the remaining portion of cameras for each dataset after adpoting our new graph preprocessing method.

| Data | | | LongSync-Naive | | | MultiSync | | | IRLS | | | MPLS on full dataset | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | $n$ | $k$ | $\bar{e}$ | $\hat{e}$ | $t$ | $\bar{e}$ | $\hat{e}$ | $t$ | $\bar{e}$ | $\hat{e}$ | $t$ | $\bar{e}$ | $\hat{e}$ | $t$ |
| Alamo | 627 | 10 | **7.45** | **1.11** | 8.91 | 7.74 | 1.56 | 81.73 | 7.56 | 1.30 | 5.71 | 3.67 | 1.02 | 55.43 |
| Ellis Island | 247 | 7 | **3.84** | **0.69** | 2.28 | 5.24 | 2.29 | 25.49 | 4.12 | 1.09 | 1.77 | 2.82 | 0.50 | 5.72 |
| Madrid Metropolis | 394 | 6 | **9.85** | **2.92** | 2.96 | 10.27 | 3.91 | 15.46 | 10.13 | 3.61 | 2.24 | 5.83 | 1.31 | 7.63 |
| Montreal Notre Dame | 474 | 8 | **5.93** | **0.61** | 5.17 | 6.49 | 1.44 | 41.20 | 6.06 | 0.87 | 3.37 | 1.13 | 0.50 | 18.40 |
| Notre Dame | 553 | 11 | **4.57** | **0.72** | 8.56 | 4.97 | 1.28 | 117.81 | 4.82 | 1.09 | 5.19 | 2.71 | 0.64 | 57.94 |
| NYC Library | 376 | 6 | **6.15** | **1.65** | 2.77 | 7.13 | 2.96 | 15.58 | 6.28 | 1.92 | 2.07 | 3.11 | 1.30 | 5.92 |
| Piazza del Popolo | 345 | 7 | **6.37** | **0.99** | 2.84 | 10.18 | 7.09 | 33.35 | 7.23 | 1.18 | 2.07 | 3.44 | 0.86 | 7.19 |
| Roman Forum | 1102 | 6 | **5.98** | **1.80** | 10.15 | 6.61 | 2.57 | 19.17 | 6.06 | 1.93 | 5.81 | 2.87 | 1.41 | 27.33 |
| Tower of London | 489 | 5 | **6.46** | **2.95** | 3.99 | 7.03 | 3.43 | 9.77 | 6.74 | 3.24 | 2.77 | 3.96 | 2.44 | 6.53 |
| Union Square | 930 | 4 | 25.68 | **5.68** | 5.95 | 27.64 | 7.24 | 7.34 | **25.31** | 5.74 | 4.20 | 6.14 | 3.70 | 8.52 |
| Vienna Cathedral | 918 | 9 | **13.26** | **1.60** | 10.31 | 13.74 | 2.37 | 62.84 | 13.47 | 1.97 | 6.33 | 6.19 | 1.31 | 58.05 |
| Gendarmenmarkt | 742 | 6 | **74.25** | 72.34 | 6.40 | 74.63 | **71.53** | 17.56 | 76.58 | 81.32 | 4.33 | 39.70 | 10.48 | 17.13 |
| Piccadilly | 2508 | 9 | **9.58** | **2.82** | 42.17 | 9.91 | 3.19 | 72.09 | 10.66 | 3.78 | 19.39 | 4.45 | 2.08 | 455.83 |
| Trafalgar | 5433 | 9 | **9.61** | **3.27** | 130.55 | 10.23 | 4.16 | 112.62 | 9.75 | 3.44 | 60.49 | 5.49 | 4.39 | 1929.21 |
| Yorkminster | 458 | 6 | **8.25** | **1.69** | 5.06 | 8.80 | 2.47 | 16.59 | 8.28 | 1.74 | 3.87 | 3.55 | 1.58 | 8.31 |

Table 4. Results for PhotoTourism where all methods are performed without our graph preprocessing method. For each dataset, $\bar{e}$ and $\hat{e}$ indicate the mean error and median error of the output absolute rotation estimates measured in degrees, and $t$ is the total runtime of each method measured in seconds. The last column indicates the remaining portion of cameras for each dataset after the camera pruning step of our improved pipeline.

## B. Proof for the Formulas of $g_c$ and $f_c$ and their Computation Complexity

In this section we prove the formulas and time complexity for $f_c$ and $g_c$ defined in section 3.

For $c=3$, since all 3-cycles are simple, $f_c(\boldsymbol{W})(i,j)=\sum_{L\in C_{ij}^c}\prod_{e\in L\setminus\{ij\}}=\sum_{k\in[n]}w_{ik}w_{kj}$ is exactly the $ij$-th entry of $\boldsymbol{W}^2$, and $g_c(\boldsymbol{W},\boldsymbol{R})(i,j)=\sum_{L\in C_{ij}^c}=\sum_{k\in[n]}w_{ik}\boldsymbol{R}_{ik}w_{kj}\boldsymbol{R}_{kj}$ is exactly the $ij$-th block of $\boldsymbol{P}^2$.

For $c\geq 4$, there are redundant cycles in $C_{ij}^c$, i.e. cycles that are not simple. We follow the argument in [35] to compute $f_c(\boldsymbol{W})(i,j)$ and $g_c(\boldsymbol{W},\boldsymbol{R})(i,j)$. For example, the cycle $ikij$ is redundant since the node $i$ repeats twice. We say this cycle satisfy the partition $0+2+1$ of $c-1$, in that the number of steps from the first node to the repeated node is 0, the number of steps from the repeated node to its second appearance is 2, and the number of remaining steps to the last letter is 1. Some cycles may satisfy more than 1 partition. For integer $1\leq a\leq c-1$, let $C_{ij,a}^c$ be the set of redundant $c$-cycles satisfying $a$ partitions. Let $q_c$ be the number of admissible partitions of length $c$, i.e. partitions that correspond to a redundant cycle. Then the function $f_c$ and $g_c$ can be written as follows:

$$f_c(\boldsymbol{W})(i,j)=\boldsymbol{W}^{c-1}+\sum_{a=1}^{q_c}(-1)^a\sum_{L\in C_{ij,a}^c}\prod_{e\in L\setminus\{ij\}}w_e \tag{15}$$

$$g_c(\boldsymbol{W},\boldsymbol{R})(i,j)=\boldsymbol{P}^{c-1}+\sum_{a=1}^{q_c}(-1)^a\sum_{L\in C_{ij,a}^c}\prod_{e\in L\setminus\{ij\}}w_e\boldsymbol{R}_e. \tag{16}$$

For $c=4$, the set of admissible partitions is $\{0+2+1,1+2+0\}$, therefore $q_4=2$. By enumerating the possible cycles for any combination of such admissible partitions, we know that the set $C_{ij,1}^4=\{k\in[n]:ikij\}\cup\{k\in[n]:ijkj\}$, and the set $C_{ij,2}^4=\{ijij\}$. Therefore we can simplify the above formulation as:

$$f_c(\boldsymbol{W})(i,j)=\boldsymbol{W}^{c-1}-\sum_{k\in[n]}w_{ik}w_{ki}w_{ij}-\sum_{k\in[n]}w_{ij}w_{jk}w_{kj}+w_{ij}w_{ji}w_{ij} \tag{17}$$

$$g_c(\boldsymbol{W},\boldsymbol{R})(i,j)=\boldsymbol{P}^{c-1}-\sum_{k\in[n]}w_{ik}w_{ki}w_{ij}\boldsymbol{R}_{ik}\boldsymbol{R}_{ki}\boldsymbol{R}_{ij}-\sum_{k\in[n]}w_{ij}w_{jk}w_{kj}\boldsymbol{R}_{ij}\boldsymbol{R}_{jk}\boldsymbol{R}_{kj}+w_{ij}w_{ji}w_{ij}\boldsymbol{R}_{ij}\boldsymbol{R}_{ji}\boldsymbol{R}_{ij}. \tag{18}$$

This can be vectorized as

$$f_c(\boldsymbol{W})=\boldsymbol{W}^3-\mathrm{d}(\boldsymbol{W}^2)\boldsymbol{W}-\boldsymbol{W}\mathrm{d}(\boldsymbol{W}^2)+\boldsymbol{W}^{\odot 3} \tag{19}$$

$$g_c(\boldsymbol{W},\boldsymbol{R})=\boldsymbol{P}^3-\mathrm{d}(\boldsymbol{P}^2)\boldsymbol{P}-\boldsymbol{P}\mathrm{d}(\boldsymbol{P}^2)+\boldsymbol{P}^{\odot 3}. \tag{20}$$

Using similar arguments as above (one may refer to [35]), we have the formulas for $c=5$ and $c=6$. The formulas for $c=5$ are presented in Table 1. The formulas for $c=6$ are as follows:

$$\begin{aligned}
f_c(\boldsymbol{W})=&\boldsymbol{W}\mathrm{d}(\boldsymbol{W}^4)+\mathrm{d}(\boldsymbol{W}^4)\boldsymbol{W}+\boldsymbol{W}^2\mathrm{d}(\boldsymbol{W}^3)+\mathrm{d}(\boldsymbol{W}^3)\boldsymbol{W}^2+\boldsymbol{W}\mathrm{d}(\boldsymbol{W}^2)\boldsymbol{W}^2+\boldsymbol{W}^2\mathrm{d}(\boldsymbol{W}^2)\boldsymbol{W}+\boldsymbol{W}\mathrm{d}(\boldsymbol{W}^3)\boldsymbol{W}\\
&+\boldsymbol{W}^2\odot\boldsymbol{W}^{\odot 3}+3\boldsymbol{W}\odot(\boldsymbol{W}^{\odot 2})^2+2\boldsymbol{W}\mathrm{d}(\boldsymbol{W}^2)\odot\boldsymbol{W}^{\odot 2}+2\mathrm{d}(\boldsymbol{W}^2)\boldsymbol{W}\odot\boldsymbol{W}^{\odot 2}\\
&+4\mathrm{d}(\boldsymbol{W}^2)\boldsymbol{W}^{\odot 3}+4\boldsymbol{W}^{\odot 3}\mathrm{d}(\boldsymbol{W}^2)-\boldsymbol{W}\mathrm{d}(\boldsymbol{W}\mathrm{d}(\boldsymbol{W}^2)\boldsymbol{W})-\mathrm{d}(\boldsymbol{W}\mathrm{d}(\boldsymbol{W}^2)\boldsymbol{W})\boldsymbol{W}\\
&-2\boldsymbol{W}(\boldsymbol{W}^{\odot 2}\odot\boldsymbol{W}^2)-2(\boldsymbol{W}^{\odot 2}\odot\boldsymbol{W}^2)\boldsymbol{W}-\boldsymbol{W}^{\odot 2}\boldsymbol{W}^2-\boldsymbol{W}^2\boldsymbol{W}^{\odot 2}\\
&-2\boldsymbol{W}\mathrm{d}(\boldsymbol{W}^2)^2-2\mathrm{d}(\boldsymbol{W}^2)^2\boldsymbol{W}-\boldsymbol{W}(\boldsymbol{W}\odot\boldsymbol{W}^2)-(\boldsymbol{W}\odot\boldsymbol{W}^2)\boldsymbol{W}-\boldsymbol{W}\odot\boldsymbol{W}^3-2\boldsymbol{W}^{\odot 2}\boldsymbol{W}^3-\mathrm{d}(\boldsymbol{W})^2\boldsymbol{W}\mathrm{d}(\boldsymbol{W}^2)\\
&-\boldsymbol{W}\odot\boldsymbol{W}^2\odot\boldsymbol{W}^2-\boldsymbol{W}\boldsymbol{W}^{\odot 3}\boldsymbol{W}-2\boldsymbol{W}\odot\boldsymbol{W}^2\odot\boldsymbol{W}^2-4\boldsymbol{W}^{\odot 5}\\
g_c(\boldsymbol{W},\boldsymbol{R})=&\boldsymbol{P}\mathrm{d}(\boldsymbol{P}^4)+\mathrm{d}(\boldsymbol{P}^4)\boldsymbol{P}+\boldsymbol{P}^2\mathrm{d}(\boldsymbol{P}^3)+\mathrm{d}(\boldsymbol{P}^3)\boldsymbol{P}^2+\boldsymbol{P}\mathrm{d}(\boldsymbol{P}^2)\boldsymbol{P}^2+\boldsymbol{P}^2\mathrm{d}(\boldsymbol{P}^2)\boldsymbol{P}+\boldsymbol{P}\mathrm{d}(\boldsymbol{P}^3)\boldsymbol{P}\\
&+\boldsymbol{P}^2\odot\boldsymbol{P}^{\odot 3}+3\boldsymbol{P}\odot(\boldsymbol{P}^{\odot 2})^2+2\boldsymbol{P}\mathrm{d}(\boldsymbol{P}^2)\odot\boldsymbol{P}^{\odot 2}+2\mathrm{d}(\boldsymbol{P}^2)\boldsymbol{P}\odot\boldsymbol{P}^{\odot 2}\\
&+4\mathrm{d}(\boldsymbol{P}^2)\boldsymbol{P}^{\odot 3}+4\boldsymbol{P}^{\odot 3}\mathrm{d}(\boldsymbol{P}^2)-\boldsymbol{P}\mathrm{d}(\boldsymbol{P}\mathrm{d}(\boldsymbol{P}^2)\boldsymbol{P})-\mathrm{d}(\boldsymbol{P}\mathrm{d}(\boldsymbol{P}^2)\boldsymbol{P})\boldsymbol{P}\\
&-2\boldsymbol{P}(\boldsymbol{P}^{\odot 2}\odot\boldsymbol{P}^2)-2(\boldsymbol{P}^{\odot 2}\odot\boldsymbol{P}^2)\boldsymbol{P}-\boldsymbol{P}^{\odot 2}\boldsymbol{P}^2-\boldsymbol{P}^2\boldsymbol{P}^{\odot 2}\\
&-2\boldsymbol{P}\mathrm{d}(\boldsymbol{P}^2)^2-2\mathrm{d}(\boldsymbol{P}^2)^2\boldsymbol{P}-\boldsymbol{P}(\boldsymbol{P}\odot\boldsymbol{P}^2)-(\boldsymbol{P}\odot\boldsymbol{P}^2)\boldsymbol{P}-\boldsymbol{P}\odot\boldsymbol{P}^3-2\boldsymbol{P}^{\odot 2}\boldsymbol{P}^3-\mathrm{d}(\boldsymbol{P})^2\boldsymbol{P}\mathrm{d}(\boldsymbol{P}^2)\\
&-\boldsymbol{P}\odot\boldsymbol{P}^2\odot\boldsymbol{P}^2-\boldsymbol{P}\boldsymbol{P}^{\odot 3}\boldsymbol{P}-2\boldsymbol{P}\odot\boldsymbol{P}^2\odot\boldsymbol{P}^2-4\boldsymbol{P}^{\odot 5}
\end{aligned}$$

The computational time complexity of the previous cases for $f_c$ and $g_c$ are $O(r(n))$ and $O(r(dn))$, respectively, since computing $f_c$ by the formula above only requires standard matrix operations between $n\times n$ matrices, and computing $g_c$ by the formula above only requires standard matrix operations between $dn\times dn$ matrices. For the case $c\geq 7$, [47] gives an estimation on the upper bound of the computational time complexity as $O(n^{\lceil (c+3)/2\rceil})$.

## C. Main Theory

We formulate theory for adversarial corruption in Section C.1 and for the uniform corruption model in Section C.2. The latter theory extends the one stated in Section 4.

Both settings use the following common notation. Let $E_g$ be the set of good (clean) edges, $E_b$ be the set of bad (corrupted) edges, and $N_{ij}^c$ be the set of simple $c$-cycles containing $ij$. Let $G_{ij}^c$ be the set of good simple $c$-cycles with respect to $ij$. That is, for any cycle $L\in G_{ij}^c$, $L$ is simple of length $c$ and $L\setminus\{ij\}$ are all clean.

### C.1. Theory for Adversarial Corruption

In this section we focus on the adversarial corruption model [25]. The adversarial corruption model makes no assumption on the graph topology or the corrution pattern. The only assumption is that for each $ij\in E_g$, $g_{ij}=g_{ij}^*$, and for each $ij\in E_b$, $g_{ij}\neq g_{ij}^*$. Since LongSync is a modified and vectorized version of CEMP for higher-order cycles, it inherits the robustness of CEMP to adversarial corruption. Define $\lambda=\max_{ij\in E}|B_{ij}^c|/|N_{ij}^c|$ where $B_{ij}^c=N_{ij}^c\setminus G_{ij}^c$ is the set of bad cycles with respect to $ij$ (namely at least one of the other $(c-1)$ edges in the cycle are corrupted). In the scenario of adversarial corruption with an assumption on $\lambda$, we can guarantee linear convergence of LongSync as follows.

**Theorem C.1.** *Assume data is generated by the adversarial corruption model with* $\lambda<\frac{1}{1+(c-1)^2}$. *Assume the parameters* $\{\beta_t\}_{t=1}^{t_{max}}$ *of LongSync with c-cycles satisfy* $\beta_0\leq 1/(c-1)$, $\beta_{t+1}=r\beta_t$ *and* $1<r<\frac{1}{c-1}\sqrt{\frac{1-\lambda}{\lambda}}$. *Then the corruption levels* $\{s_{ij}^{(t)}\}_{ij\in E}$ *estimated by LongSync satisfy the following equation:*

$$\max_{ij\in E}|s_{ij}^{(t)}-s_{ij}^*|\leq\frac{1}{(c-1)\beta_0 r^t}\ \textit{for all}\ t\geq 0. \tag{21}$$

*Proof.* Let $\epsilon_{ij}(t)=|s_{ij}^{(t)}-s_{ij}^*|$ and $\epsilon(t)=\max_{ij\in E}\epsilon_{ij}(t)$. By the fact that $|d_L-s_{ij}^*|\leq s_L^*$, $G_{ij}^c\subseteq N_{ij}^c$ and $s_L^*=0$ for $L\in G_{ij}^c$, we obtain that

$$\begin{aligned}(\epsilon_{ij}(t+1))^2=|s_{ij}^{(t)}-s_{ij}^*|^2&=\left|\sqrt{\frac{\sum_{L\in N_{ij}^c}e^{-\beta_t s_L^{(t)}}d_L^2}{\sum_{L\in N_{ij}^c}e^{-\beta_t s_L^{(t)}}}}-s_{ij}^*\right|^2\\
&\leq\frac{\sum_{L\in N_{ij}^c}e^{-\beta_t s_L^{(t)}}|d_L-s_{ij}^*|^2}{\sum_{L\in N_{ij}^c}e^{-\beta_t s_L^{(t)}}}\\
&\leq\frac{\sum_{L\in N_{ij}^c}e^{-\beta_t s_L^{(t)}}(s_L^*)^2}{\sum_{L\in N_{ij}^c}e^{-\beta_t s_L^{(t)}}}\\
&\leq\frac{\sum_{L\in B_{ij}^c}e^{-\beta_t s_L^{(t)}}(s_L^*)^2}{\sum_{L\in G_{ij}^c}e^{-\beta_t s_L^{(t)}}}\\
&\leq\frac{\sum_{L\in B_{ij}^c}e^{-\beta_t\sum_{e\in L}\epsilon_e(t)}(s_L^*)^2}{\sum_{L\in G_{ij}^c}e^{-\beta_t\sum_{e\in L}\epsilon_e(t)}}\\
&\leq\frac{1}{|G_{ij}^c|}e^{2\beta_t(c-1)\epsilon(t)}\sum_{L\in B_{ij}^c}e^{-\beta_t s_L^*}(s_L^*)^2.\end{aligned} \tag{22}$$

We prove the theorem by induction. Note that the case $t=0$ is equivalent to $\epsilon(0)\leq 1/(c-1)\beta_0$, and this immediately follows from the fact that $0\leq\epsilon_{ij}(0)\leq 1$ and the assumption $\beta_0<1/(c-1)$. We next prove $\epsilon(t+1)<1/(c-1)\beta_{t+1}$ from $\epsilon(t)<1/(c-1)\beta_t$. By the inequality above, the induction assumption, the fact that $x^2e^x < 4/(ax)^2$ with $x = s_L^*$ and $a = \beta_t$ and the definition of $\lambda$ and $r$ we have

$$(\epsilon_{ij}(t+1))^2 \le \frac{1}{|G_{ij}^c|}\cdot e^2 \cdot \frac{4|B_{ij}^c|}{e^2\beta_t^2} = \frac{4|B_{ij}^c|}{|G_{ij}^c|\beta_t^2} \le \frac{4\lambda}{(1-\lambda)\beta_t^2} = \frac{1}{\beta_t^2 r^2(c-1)^2} = \frac{1}{\beta_{t+1}^2(c-1)^2}. \tag{23}$$

The theorem follows by taking the maximum of the left hand side and then the square root of both sides of the above equation. □

### C.2. Theory for Uniform Corruption Model

Throughout the rest of the paper we use $P(A)$ to denote the probability of event $A$. Let $p_0 = P(g_{ij} = g_{ij}^*)$ for each edge $ij \in E_b$. By the choice of corruption model, $p_0$ only depends on the group $\mathcal{G}$. Let $q_* = 1-q+qp_0 = P(ij \in E_g | ij \in E)$. Let $q_g = 1-q$. We remark that for rotation synchronization (in fact any Lie group synchronization), $q_g = q_*$ and $p_0 = 0$.

Recall for each $e \in E$, $s_e^*$ is the ground truth corruption level of edge $e$. For $L = (ik_1, k_1k_2, \cdots, k_{c-2}j) \in N_{ij}^c$, we denote $s_L^* = \sum_{e \in L\setminus\{ij\}} s_e^*$. To state our main theorem, we let $\mathcal{F}(\beta) = \{f_\tau(x) := e^{-\tau x + 2}\tau^2x^2/4 : \tau > \beta\}$ and $V(\beta) = \sup_{\tau > \beta}\mathrm{Var}(f_\tau(s_L^*))$. Due to the model assumptions, the distribution of $f_\tau(s_L^*)$ is independent of the choice of $L \in N_{ij}^c$.

Using the above notation, we formulate the following theorem, which generalizes Theorem 4.1

**Theorem C.2.** *Let $0 < r < 1$, $0 < q < 1$, $0 < p \le 1$. Assume we use LongSync with cycles of length $c$ and $n/\log n = \Omega((pq_g)^{-\frac{c-1}{c-2-\epsilon}})$ for some $\epsilon > 0$. Assume*

$$0 < \frac{1}{\beta_0} < \frac{q_g^{c-1}q_*^{c-1}}{16(1-q_*^{c-1})(c-1)^2\beta_1}, \tag{24}$$

$$V(\beta_1) < \frac{r}{16(c-1)}\cdot\frac{q_*^{c-1}}{1-q_*^{c-1}}, \tag{25}$$

$$1/\beta_{t+1} = r/\beta_t \text{ for all } t \ge 1, \tag{26}$$

$$\min(np, n^{c-2-\epsilon}p^{c-1}) \gtrsim \frac{(1-q_*^{c-1})^2}{q_*^{2(c-1)}r^2}. \tag{27}$$

*Then with probability at least* $1 - 4cn^2\exp\left(-K\eta_0^2(pq_*)^{\frac{c-1}{c-2}}n\right) - 2e^2c\cdot\exp\left(-n^{\epsilon/(c-1)} + c\log n\right) - n^2\exp\left(-\frac{\ln 2}{2}\min(np, n^{c-2-\epsilon}p^{c-1})V(\beta_1)\right) - 2n^2\cdot\exp\left(-\frac{\eta e_{\mathcal{G}}}{8c}\ln(1+\frac{e_{\mathcal{G}}}{2(c-1)\beta_0 v_{\mathcal{G}}})\min(np, n^{c-2-\epsilon}p^{c-1})\right)$, *where $\eta_0, \eta, K, e_{\mathcal{G}}, v_{\mathcal{G}}$ are absolute constants, we have* $\max_{ij \in E}|s_{ij}^* - s_{ij}^{(t)}| \le \frac{1}{2c\beta_t}$ *for all $t \ge 1$.*

*Remark* C.3. As is shown in [25], for $\mathcal{G} \in SO(3)$, $V(\beta) \sim O(\beta^{-3})$. Therefore $n/\log n \sim p^{-(c-1)/(c-2-\epsilon)}q_g^{-7(c-1)/3(c-2-\epsilon)}$ is the minimal sample complexity dependence for $\mathcal{G} = SO(3)$ such that with high probability, the conclusion of Theorem 4.1 holds true.

### C.3. Proof of Theorem C.2

We adopt the proof framework of [25]. The major difficulty of the proof is the dependence in the cycle inconsistency measures of cycles in $N_{ij}^c$ when $c \ge 4$. For example, the cycle inconsistency measure of a 4-cycle $L_1 = (ik_1, k_1k_2, k_2j)$ is not independent with that of $L_2 = (ik_1, k_1k_3, k_3j)$, while for a pair of 3-cycles their ratios are always independent. This means that the required concentration inequalities cannot be obtained by directly applying the standard Chernoff bounds. Nonetheless, we have integrated various mathematical techniques from [3, 7, 22–24, 48] to derive Theorem 4.1, which offers improvements over theorem 7 presented in [25].

For convenience for any $c \ge 3$, we define a $c$-path as a path that involves $c$ vertices, and we define an $ij,c$-path as a $c$-path that starts from $i$ and ends at $j$. We extend the definition of $N_{ij}^c$ as the set of $ij,c$-paths in graph $G$.

We first prove that with high probability, the number of $c_1$-cycles concentrates around its mean for any $c_1 \le c$. More specifically, let $n_{c_1} = (n-2)(n-3)(n-4)\cdots(n-c_1+1)$ be the number of possible $ij,c_1$-path candidates, and $m_{c_1} = \max(p^{c_1-1}n_{c_1}, n^\epsilon)$. Therefore the expected number of $ij,c_1$-paths is $p^{c_1-1}n_{c_1}$. For any $\epsilon, \eta > 0$ we define the $(\epsilon, \eta_0)$-regular Erdős-Rényi graph condition as follows:

**Definition C.4.** Let $\delta = \sup\{\delta > 0 \text{ s.t. } np^{1+\delta}/\log n \to \infty\}$ and $c_0 = \lceil 2+\delta^{-1}\rceil$. A graph $G$ satisfies the $(\epsilon, \eta_0)$-regular Erdős-Rényi graph condition if and only if the following conditions hold true:

- For any $i \ne j \in [n]$ and $c_1 \ge c_0$,

$$(1-\eta_0)m_{c_1} < |N_{ij}^{c_1}| < (1+\eta_0)m_{c_1} \tag{28}$$

and

$$(1-\eta_0)q_*^{c_1-1}m_{c_1} < |G_{ij}^{c_1}| < (1+\eta_0)q_*^{c_1-1}m_{c_1}; \tag{29}$$

- For any $i \neq j \in [n]$ and $c_1 < c_0$,

$$0 \leq |N_{ij}^{c_1}| < m_{c_1}. \tag{30}$$

.

We have the following theorem on the phase transition of the number of $c$-paths:

**Theorem C.5.** *Assume $G$ is generated with the uniform corruption model UCM($n,p,q$), and $\epsilon,\eta > 0$ are constants. Then the $(\epsilon,\eta_0)$-regular E-R graph condition holds with probability at least $1 - cn^2\exp(-\frac{\eta_0^2}{5c}pn) - cn^2\exp(-K\eta_0^2 p^{\frac{c-1}{c-2}}n) - cn^2\exp(-\frac{\eta_0^2}{5c}pq_* n) - cn^2\exp\left(-K\eta_0^2 (pq_*)^{\frac{c-1}{c-2}}n\right) - 2e^2cn^2\exp\left(-n^{\epsilon/(c-1)} + (c-2)\log n\right)$, which is almost 1 by the condition $n/\log n = \Omega((pq_g)^{-\frac{c-1}{c-2-\epsilon}})$.*

The proof of Theorem C.5 is put in section D. Based on this theorem, we have a concentrated 'initialization' of corruption level estimates after the first iteration:

**Theorem C.6.** *(Initialization) Assume the $(\epsilon,\eta_0)$-regular E-R graph condition holds. Recall that the corruption level estimation of LongSync with cycle length $c$ at $t = 0$ is*

$$s_{ij}^{(0)} = \sqrt{\frac{\sum_{L \in N_{ij}^c} d_L^2}{|N_{ij}^c|}}. \tag{31}$$

*Denote $e_{\mathcal{G}} = \mathbb{E}d_L^2$ and $v_{\mathcal{G}} = Var(d_L^2)$. Then for any $\eta > 0$ and $ij \in E$,*

$$P(|(s_{ij}^{(0)})^2 - \mathbb{E}(s_{ij}^{(0)})^2| > \eta\mathbb{E}(s_{ij}^{(0)})^2) < 2\exp\left(-\frac{\eta e_{\mathcal{G}}}{8c}\ln(1 + \frac{\eta e_{\mathcal{G}}}{2v_{\mathcal{G}}})\min(np, n^{c-2-\epsilon}p^{c-1})\right). \tag{32}$$

Let $\lambda = \max_{ij \in E}|B_{ij}^c|/|N_{ij}^c|$ where $B_{ij}^c = N_{ij}^c \setminus G_{ij}^c$ is the set of bad $ij,c$-paths. To prove the linear convergence, we need the following three lemmas:

*Lemma* C.7. If $\max_{ij \in E}|(s_{ij}^{(0)})^2 - \mathbb{E}(s_{ij}^{(0)})^2| \leq \frac{1}{2(c-1)\beta_0}$, then

$$\max_{ij \in E}|s_{ij}^{(1)} - s_{ij}^*| \leq \frac{\lambda}{1-\lambda}\frac{2(c-1)}{q_g^{c-1}\beta_0}. \tag{33}$$

*Lemma* C.8. Assume that $\max_{ij \in E}|s_{ij}^{(1)} - s_{ij}^*| < 1/(2(c-1)\beta_1)$, $\beta_t = r\beta_{t+1}$ for $t \geq 1$, and

$$\max_{ij \in E}\frac{1}{|B_{ij}^c|}\sum_{L \in B_{ij,c}} e^{-\beta_t s_L^*}(s_L^*)^2 < \frac{1}{M\beta_t^2} \text{ for all } t \geq 1, \tag{34}$$

where $M = 4(c-1)^2 e\lambda/((1-\lambda)r^2)$. Then the LongSync corruption level estimates satisfy

$$\max_{ij \in E}|s_{ij}^{(t)} - s_{ij}^*| < \frac{1}{\beta_1}r^{t-1} \text{ for all } t \geq 1. \tag{35}$$

*Lemma* C.9. If either $s_{ij}^*$ for $ij \in E_b$ is supported on $[a,\infty)$ and $a \geq 1/|B_{ij}^c|$ or $Q$ is differentiable and $Q'(x)/Q(x) \lesssim 1/x$ for $x < P(1)$, then there exists an absolute constant $K''$ such that

$$\begin{aligned}
&P\left(\sup_{f_\tau \in \mathcal{F}(\beta)}\frac{1}{|B_{ij}^c|}\sum_{L \in B_{ij}^c} f_\tau(s_L^*) > V(\beta)\right. \\
&\left. + K''\sqrt{\frac{\log\min(np, n^{c-2-\epsilon}p^{c-1})}{\min(np, n^{c-2-\epsilon}p^{c-1})}}\right) \\
&< \exp\left(-\frac{\ln 2}{2}\min(np, n^{c-2-\epsilon}p^{c-1})V(\beta)\right).
\end{aligned} \tag{36}$$

where $\mathcal{F}(\beta) = \{f_\tau(x) = e^{-\tau x + 2}\tau^2 x^2/4 : \tau > \beta\}$.

Lemma C.7 and C.8 are direct extensions of lemma 4 and lemma 5 of [25]. Lemma C.9, however, involves the extension of theorem 2.3 in [3] to the supremum of locally independent empirical processes and Hajnal-Szemerédi theorem for equitable coloring. We refer the reader to section D for the proof of these lemmas.

*Proof of the main theorem.* By the regular E-R graph condition, we can choose appropriate $\eta_0$ so that

$$\frac{1}{4}\frac{q_*^{c-1}}{1-q_*^{c-1}} < \frac{1-\lambda}{\lambda} < 4\frac{q_*^{c-1}}{1-q_*^{c-1}}. \quad (37)$$

To guarantee the condition (34) of Lemma C.8, we need to choose $\beta_1$ such that $V(\beta_1) < e/2M$ and $n$ large enough such that $\log(\min(np,\ n^{c-2-\epsilon}p^{c-1}))/\ \min(np,\ n^{c-2-\epsilon}p^{c-1}) < e^2/4K''^2M^2$. By the assumption that $V(\beta_1) < (rq_*^{c-1})/16(c-1)(1-q_*^{c-1})$, $M = 4(c-1)^2e\lambda/((1-\lambda)r^2)$ and (37) we know that $V(\beta_1) < e/2M$. By the assumption that $\min(np, n^{c-2-\epsilon}p^{c-1}) \gtrsim (1-q_*^{c-1})^2/q_*^{2(c-1)}r^2$ we know that $\log(\min(np, n^{c-2-\epsilon}p^{c-1}))/\min(np, n^{c-2-\epsilon}p^{c-1}) < e^2/4K''^2M^2$. Therefore the condition (34) of Lemma C.8 holds true.

On the other hand, by Theorem C.6 with $\eta = 1/2(c-1)\beta_0$ we know that w.h.p. the condition of Lemma C.7 holds true. By the assumption that $1/\beta_0 < q_*^{c-1}q_g^{c-1}/16(1-q_*^{c-1})(c-1)^2\beta_1$, we know that the conclusion of Lemma C.7 implies the first assumption of Lemma C.8.

Therefore, the proof of the theorem follows from the conclusion of Lemma C.8. □

## D. Proofs of Auxiliary Results

We provide additional results for auxiliary theorems and lemmata used in the previous section.

*Proof of Theorem C.5.* We have the following basic lemmas:

*Lemma* D.1. (Concentration of number of paths of length $\geq c_0 - 1$ with fixed endpoints) Let $0 \leq q < 1$, $0 < p \leq 1$, $n \in \mathbb{N}$ with $np \geq \Theta(1)$. Assume data is generated by UCM(n,p,q), and $c \geq c_0$. For any $\eta_0 > 0$, there exists a constant $K > 0$ that only depends on $c$, such that

$$P(|N_{ij}^c| - p^{c-1}n_c < \eta_0 p^{c-1}n_c) < \exp(-\frac{\eta_0^2}{5c}pn) \quad (38)$$

$$P(|N_{ij}^c| - p^{c-1}n_c > \eta_0 p^{c-1}n_c) < \exp(-K\eta_0^2 p^{\frac{c-1}{c-2}}n) \quad (39)$$

for any fixed $i \neq j \in V$, and

$$P(|N_{ij}^c| - p^{c-1}n_c < \eta_0 p^{c-1}n_c) < |E|\exp(-\frac{\eta_0^2}{5c}pn) \quad (40)$$

$$P(|N_{ij}^c| - p^{c-1}n_c > \eta_0 p^{c-1}n_c) < |E|\exp(-K\eta_0^2 p^{\frac{c-1}{c-2}}n). \quad (41)$$

*Proof.* Let $M_{ij}^c = \{(i,k_1,k_2,\cdots,k_{c-2},j) : i,k_1,k_2,\cdots,k_{c-2},j \in [n] \text{ are different}\}$. Note that $|N_{ij}^c| = \sum_{\alpha\in M_{ij}^c} I_\alpha$, where $I_\alpha = 1_{ik_1\in E}1_{k_1k_2\in E}\cdots 1_{k_{c-3}k_{c-2}\in E}1_{k_{c-2}j\in E}$ for $\alpha = (i,k_1,k_2,\cdots,k_{c-2},j)$. For any $\alpha,\beta \in M_{ij}^c$, define $\omega = \sum_{\alpha\in M_{ij}^c}\mathbb{E}I_\alpha = \sum_{\alpha\in M_{ij}^c}p^{c-1} = p^{c-1}n_c$. Let us write $\alpha\sim\beta$ if $\alpha,\ \beta\in M_{ij}^c$ with at least one common edge, and define $\delta = (\sum_{\alpha\sim\beta}\mathbb{E}I_\alpha I_\beta)/\omega$. (This sum should be interpreted as the sum over all pairs $(\alpha,\beta)$, so each pair is counted twice.) By theorem 1 of [22], we have the following inequality:

$$P(|N_{ij}^c| < (1-\eta_0)p^{c-1}n_c) \leq \exp(-\frac{\eta_0^2\omega}{2(1+\delta)}). \quad (42)$$

Denote $|\alpha\backslash\beta|$ as the number of nodes that belong to $\beta$ but do not belong to $\alpha$. By the definition of $\delta$, we have the following estimate:

$$
\begin{aligned}
\delta &= (\sum_{\alpha\sim\beta} \mathbb{E} I_\alpha I_\beta)/\omega \\
&= \frac{1}{\omega} \sum_{\alpha\in M^c_{ij}} \sum_{k=1}^{c-3} \sum_{\alpha\sim\beta \text{ and } |\alpha\backslash\beta|=k} \mathbb{E} I_\alpha I_\beta \\
&= \frac{|M^c_{ij}|}{\omega} \sum_{k=1}^{c-3} \sum_{\alpha\sim\beta \text{ and } |\alpha\backslash\beta|=k} p^{k+c-1} \\
&\leq \frac{(n-2)(n-3)\cdots(n-c+1)}{p^{c-1}(n-2)(n-3)\cdots(n-c+1)} \sum_{k=1}^{c-3} (n-2)(n-3)\cdots(n-k-1)p^{k+c} \\
&\leq \frac{1}{p^{c-1}} c(n-2)(n-3)\cdots(n-c+2)p^{2c-3} \\
&\leq c(n-2)(n-3)\cdots(n-c+2)p^{c-2} = \frac{c\omega}{(n-c+1)p}.
\end{aligned} \tag{43}
$$

Plugging (43) to (42) gives:

$$
\begin{aligned}
P(|N^c_{ij}| < (1-\eta_0)p^{c-1}n_c) &\leq \exp(-\frac{\eta_0^2 \omega}{2(1+\delta)}) \\
&< \exp(-\frac{\eta_0^2\omega}{4\delta}) \\
&\leq \exp(-\frac{\eta_0^2 \omega (n-c+1)p}{4c\omega}) \\
&< \exp(-\frac{\eta_0^2 np}{5c}).
\end{aligned} \tag{44}
$$

Therefore inequality (38) is proved, and inequality (40) follows from a union bound argument.

For the upper tail, let $A$ be an arbitrary subset of $\{k_1, k_2, \cdots, k_{c-2}\}$, the set of free vertices of an $ij,c$-path. Denote $\mathbb{M}_A$ as the expected number of $ij,c$-paths $(ik_1, k_1k_2, \cdots, k_{c-2}j)$, where the vertices in $A$ are fixed, and let $\mathbb{M}_k = \max_{|A|\geq k} \mathbb{M}_A$. We have the following calculation:

$$
\mathbb{M}_k = \begin{cases} n^{c-2-k}p^{c-1-k}, & k \leq c-3 \\ 1, & k = c-2 \end{cases}. \tag{45}
$$

Let $\lambda = \eta_0^2(n-c+1)p^{\frac{c-1}{c-2}}$. By $c \geq c_0$, we know that $\lambda = \omega(\log n)$. Also, by setting $M_0 = \mathbb{M}_0$ and $M_k = M_0 \lambda^{-k}$ we know that for all $0 \leq k \leq c-2$, $M_k \geq \mathbb{M}_k$. Therefore we can apply theorem 1.2 in [48] and get the following inequality

$$
P(|N^c_{ij}| - p^{c-1}n_c > \eta_0 n_c) \leq \exp(-K_0 \eta_0^2 (n-c+1)p^{\frac{c-1}{c-2}}) \tag{46}
$$

where $K_0$ is a constant that only depends on $c$. Let $K = K_0/2$. By the order of $c$ we know that

$$
P(|N^c_{ij}| - p^{c-1}n_c > \eta_0 n_c) \leq \exp(-K\eta_0^2 np^{\frac{c-1}{c-2}}). \tag{47}
$$

Therefore inequality (39) is proved, and inequality (41) follows from a union bound argument. □

*Lemma* D.2. Let $0 \leq q < 1$, $0 < p \leq 1$, $n \in \mathbb{N}$ with $np \geq \Theta(1)$. Assume data is generated by UCM(n,p,q), $c \geq c_0$, and $K$ is the constant in Lemma D.1. For any $\eta_0 > 0$, we have

$$
P(|G^c_{ij}| - p^{c-1}q_*^{c-1}n_c < \eta_0 p^{c-1}q_*^{c-1}n_c) < \exp(-\frac{\eta_0^2}{5c}pq_*n) \tag{48}
$$

$$
P(|G^c_{ij}| - p^{c-1}q_*^{c-1}n_c > \eta_0 p^{c-1}q_*^{c-1}n_c) < \exp(-K\eta_0^2 pq_*n) \tag{49}
$$

for any fixed $i \neq j \in V$, and

$$P(|G_{ij}^c| - p^{c-1}q_*^{c-1}n_c < \eta_0 p^{c-1}q_*^{c-1}n_c) < |E|\exp(-\frac{\eta_0^2}{5c}pq_*n) \tag{50}$$

$$P(|G_{ij}^c| - p^{c-1}q_*^{c-1}n_c > \eta_0 p^{c-1}q_*^{c-1}n_c) < |E|\exp(-K\eta_0^2 pq_*n). \tag{51}$$

Lemma D.2 is proved by replacing $p$ with $pq_*$ in the proof of Lemma D.1.

To count the shorter paths which has a vanishing expectation when $n$ tends to infinity, we need the following concentration inequality:

*Lemma* D.3. (Concentration of number of paths with length $\le c_0 - 2$) Let $0 \le q < 1$, $0 < p \le 1$, $n \in \mathbb{N}$ with $np \ge \Theta(1)$. Assume data is generated by UCM(n,p,q), and $c < c_0$. For any $\epsilon > 0$, there exists a constant $K' > 0$ that only depends on c, such that

$$P(|N_{ij}^c| > K'n^\epsilon) < 2e^2\exp(-n^{\epsilon/(c-1)} + (c-2)\log n) \tag{52}$$

for any fixed $i \neq j \in V$, and

$$P(|N_{ij}^c| > K'n^\epsilon) < 2e^2|E|\exp(-n^{\epsilon/(c-1)} + (c-2)\log n). \tag{53}$$

*Proof.* Define the multivariable polynomial $f(\{x_{pq}\}_{p\neq q\in[n]}) = \sum_{\alpha\in M_{ij}^c} x_\alpha$, where $x_\alpha = x_{ik_1}x_{k_1k_2}\cdots x_{k_{c-2}j}$ for $\alpha = (i,k_1,k_2,\cdots,k_{c-2},j)$ in $M_{ij}^c = \{(i,k_1,k_2,\cdots,k_{c-2},j) : i,k_1,k_2,\cdots,k_{c-2},j \in [n]$ are different$\}$. Note that $|N_{ij}^c| = f(\{1_{pq\in E}\}_{p\neq q\in[n]})$. Let $A \subseteq \{x_{pq\in E} : p \neq q \in [n]\}$ be a subset of the variables of $f$, and $f_A(\{x_{pq}\}_{p\neq q\in[n]})$ be the partial derivative of $f(\{x_{pq}\}_{p\neq q\in[n]})$ with respect to all variables in $A$. Let $\partial_A|N_{ij}^c| = f_A(\{1_{pq\in E}\}_{p\neq q\in[n]})$. Define $E_k = \max_{|A|\ge k}\mathbb{E}(\partial_A|N_{ij}^c|)$. By the main theorem in [24], we know that

$$P(|N_{ij}^c - E_0| > K'n^{(c-1)\epsilon}\sqrt{E_0E_1}) < 2e^2\exp(-n^\epsilon + (c-2)\log n). \tag{54}$$

Because $c < c_0$, we know that for any $k \in \mathbb{N}$, $\max_{|A|\le c-2}\mathbb{E}(\partial_A|N_{ij}|) = o(1)$ and $\max_{|A|=c-1}\mathbb{E}(\partial_A|N_{ij}|) = 1$. Therefore, $E_0 = E_1 = 1$. Plugging these values into inequality (54) and substituting $\epsilon$ with $\epsilon/(c-1)$ results in inequality (52). Inequality (53) is obtained from a union probability bound argument. □

With the estimates above, the regular E-R graph condition holds with probability at least $1 - n^2\exp(-\frac{\eta_0^2}{5c}pn) - n^2\exp(-K\eta_0^2 p^{\frac{c-1}{c-2}}n) - n^2\exp(-\frac{\eta_0^2}{5c}pq_*n) - n^2\exp(-K\eta_0^2(pq_*)^{\frac{c-1}{c-2}}n) - 2e^2n^2\exp(-n^\epsilon + (c-2)\log n)$.

□

*Proof of Theorem C.6.* For any $L \in N_{ij}^c$ and $pq \in L$, we say $L'$ is correlated with $L$ if $L \cap L'$ is nonempty, and $L'$ is correlated with $L\backslash\{pq\}$ if $(L\backslash\{pq\}) \cap L'$ is nonempty. We denote $C_L$ as the set of $ij,c$-paths in $N_{ij}^c$ that is correlated with $L$, and denote $C_{L\backslash\{pq\}}$ as the set of $ij,c$-paths in $N_{ij}^c$ that is correlated with $L\backslash\{pq\}$. With the regular E-R graph condition, we know that for any $L \in N_{ij}^c$,

$$|C_L| \le \sum_{pq\in L}|C_{L\backslash\{pq\}}| \tag{55}$$

$$\le m_{c-1} + m_1m_{c-2} + m_2m_{c-3} + \cdots + m_{c-2}m_1 + m_{c-1} \tag{56}$$

$$< cm_{c-1}. \tag{57}$$

Denote $\Delta_1 = \max_{L\in N_{ij}^c}|C_L|$. Then we know that $\Delta_1 < cm_{c-1} < c\max(n^\epsilon, n^{c-3}p^{c-2})$. We apply theorem 2.5 in [23] on $\sum_{L\in N_{ij}^c} d_L^2$ and $\sum_{L\in N_{ij}^c}(-d_L^2)$ and get the following inequalities:

$$P(\sum_{L\in N_{ij}^c} d_L^2 > (1+\eta)\mathbb{E}\sum_{L\in N_{ij}^c} d_L^2) < \exp(-\frac{|N_{ij}^c|v_G}{\Delta_1}\varphi(\frac{\eta\mathbb{E}\sum_{L\in N_{ij}^c} d_L^2}{|N_{ij}^c|v_G(1+\Delta_1/8|N_{ij}^c|)})) \tag{58}$$

and

$$P(\sum_{L\in N_{ij}^c} d_L^2 < (1-\eta)\mathbb{E}\sum_{L\in N_{ij}^c} d_L^2) < \exp(-\frac{|N_{ij}^c|v_G}{\Delta_1}\varphi(\frac{\eta\mathbb{E}\sum_{L\in N_{ij}^c} d_L^2}{|N_{ij}^c|v_G(1+\Delta_1/8|N_{ij}^c|)})) \tag{59}$$

where $\varphi(x) = (1+x)\ln(1+x) - x$. Note that $\varphi(x) \ge x\ln(1+x)/2$ for any $x \ge 0$. By the regular E-R graph condition we have $|N_{ij}^c| \ge (1-\eta_0)n^{c-2}p^{c-1}$, and therefore $\Delta_1/|N_{ij}^c| \le \max(1/(n^{c-2}p^{c-1}), 1/(np))/(1-\eta_0) < 1$. Also, since all the $d_L^2$'s for $L \in N_{ij}^c$

follow the same distribution with mean $e_{\mathcal{G}}$ and variance $v_{\mathcal{G}}$, we know that $\mathbb{E}\sum_{L\in N_{ij}^c} d_L^2 = |N_{ij}^c| e_{\mathcal{G}}$. Therefore RHS of (58) and (59) can be upper bounded as follows:

$$\begin{aligned}
\text{RHS of (58) and (59)} &\le \exp\Bigg(-\frac{|N_{ij}^c|v_{\mathcal{G}}}{\Delta_1}\cdot\frac{\eta\mathbb{E}\sum_{L\in N_{ij}^c} d_L^2}{2|N_{ij}^c|v_{\mathcal{G}}(1+\Delta_1/8|N_{ij}^c|)} \\
&\quad\cdot\ln\Big(1+\frac{\eta\mathbb{E}\sum_{L\in N_{ij}^c} d_L^2}{|N_{ij}^c|v_{\mathcal{G}}(1+\Delta_1/8|N_{ij}^c|)}\Big)\Bigg) \\
&= \exp\Bigg(-\frac{1}{\Delta_1}\cdot\frac{\eta|N_{ij}^c|e_{\mathcal{G}}}{2(1+\Delta_1/8|N_{ij}^c|)}\ln\Big(1+\frac{\eta e_{\mathcal{G}}}{v_{\mathcal{G}}(1+\Delta_1/8|N_{ij}^c|)}\Big)\Bigg) \\
&\le \exp\Bigg(-\frac{\eta e_{\mathcal{G}}|N_{ij}^c|}{4\Delta_1}\ln\Big(1+\frac{\eta e_{\mathcal{G}}}{2v_{\mathcal{G}}}\Big)\Bigg) \\
&\le \exp\Bigg(-\frac{\eta e_{\mathcal{G}}(1-\eta_0)n^{c-2}p^{c-1}}{4\max(n^\epsilon, n^{c-3}p^{c-2})}\ln\Big(1+\frac{\eta e_{\mathcal{G}}}{2v_{\mathcal{G}}}\Big)\Bigg) \\
&\le \exp\Bigg(-\frac{\eta e_{\mathcal{G}}}{8c}\ln\Big(1+\frac{\eta e_{\mathcal{G}}}{2v_{\mathcal{G}}}\Big)\min(np, n^{c-2-\epsilon}p^{c-1})\Bigg).
\end{aligned} \tag{60}$$

Combining the upper and lower tail bound together yields

$$P\Big(\Big|\sum_{L\in N_{ij}^c} d_L^2 - \mathbb{E}\sum_{L\in N_{ij}^c} d_L^2\Big| > \eta\mathbb{E}\sum_{L\in N_{ij}^c} d_L^2\Big) < 2\exp\Big(-\frac{\eta e_{\mathcal{G}}}{8c}\ln\Big(1+\frac{\eta e_{\mathcal{G}}}{2v_{\mathcal{G}}}\Big)\min(np, n^{c-2-\epsilon}p^{c-1})\Big). \tag{61}$$

Then Theorem C.6 follows by (31). □

*Proof of Lemma C.7.* Denote $\gamma_{ij} = (s_{ij}^{(0)})^2 - \mathbb{E}(s_{ij}^{(0)})^2$ for $ij\in E$ and $\gamma = \max_{ij\in E}|\gamma_{ij}|$, so that the condition of the lemma can be written more simply as $1/2(c-1)\beta_0 \ge \gamma$. By rewriting $\mathbb{E}(s_{ij}^{(0)})^2$ as $q_g^{c-1}(s_{ij}^*)^2 + (1-q_g^{c-1})z_{\mathcal{G}} + \gamma_{ij}$ and invoking lemma 1 in [25] and equations (6) (7), we have the following bound:

$$\begin{aligned}
|s_{ij}^{(1)} - s_{ij}^*|^2 &\le \frac{\sum_{L\in N_{ij}^c} e^{-\beta_0\sqrt{\sum_{e\in L} q_g^{c-1}(s_e^*)^2 + (1-q_g^{c-1})z_{\mathcal{G}} + \gamma_e}}|d_L - s_{ij}^*|^2}{\sum_{L\in N_{ij}^c} e^{-\beta_0\sum_{e\in L}\sqrt{q_g^{c-1}(s_e^*)^2 + (1-q_g^{c-1})z_{\mathcal{G}} + \gamma_e}}} \\
&\le \frac{\sum_{L\in B_{ij}^c} e^{-\beta_0\sum_{e\in L}\sqrt{q_g^{c-1}(s_e^*)^2 + (1-q_g^{c-1})z_{\mathcal{G}} + \gamma_e}}(s_L^*)^2}{\sum_{L\in G_{ij}^c} e^{-\beta_0\sum_{e\in L}\sqrt{q_g^{c-1}(s_e^*)^2 + (1-q_g^{c-1})z_{\mathcal{G}} + \gamma_e}}}
\end{aligned} \tag{62}$$

By first applying the facts: $|\gamma_e| \le \gamma$ and $s_e^* = 0$ for $e \in L$ where $L \in G_{ij}^c$, and at last the inequality $xe^{-ax} \le 1/(ea)$ with $x = \sum_{e\in L}(s_e^*)^2$ and $a = \beta_0 q_g^{c-1}/2$, we obtain that

$$\begin{aligned}
|s_{ij}^{(1)} - s_{ij}^*|^2 &\le \frac{\sum_{L\in B_{ij}^c} e^{-\beta_0\sum_{e\in L}\sqrt{q_g^{c-1}(s_e^*)^2 + (1-q_g^{c-1})z_{\mathcal{G}} - \gamma}}(s_L^*)^2}{|G_{ij}^c| e^{-\beta_0(c-1)\sqrt{(1-q_g^{c-1})z_{\mathcal{G}} + \gamma}}} \\
&= \frac{\sum_{L\in B_{ij}^c} e^{-\beta_0\sum_{e\in L}(\sqrt{q_g^{c-1}(s_e^*)^2 + (1-q_g^{c-1})z_{\mathcal{G}} - \gamma} - \sqrt{(1-q_g^{c-1})z_{\mathcal{G}} + \gamma})}(s_L^*)^2}{|G_{ij}^c|} \\
&\le \frac{\sum_{L\in B_{ij}^c} e^{-\beta_0\sum_{e\in L}(q_g^{c-1}(s_e^*)^2 - 2\gamma)/2}(s_L^*)^2}{|G_{ij}^c|} \\
&\le \frac{e^{2\beta_0(c-1)\gamma}\sum_{L\in B_{ij}^c} e^{-\beta_0 q_g^{c-1}\sum_{e\in L}(s_e^*)^2/2}(c-1)\sum_{e\in L}(s_e^*)^2}{|G_{ij}^c|} \\
&\le \frac{2(c-1)|B_{ij}^c|}{|G_{ij}^c|\beta_0 q_g^{c-1}}.
\end{aligned} \tag{63}$$

The lemma is concluded by applying the union bound on $ij \in E$ and taking the square root on both sides of the above inequality. $\square$

*Proof of Lemma C.8.* Let $\epsilon_{ij}(t) = |s_{ij}^{(t)} - s_{ij}^*|$ and $\epsilon(t) = \max_{ij\in E}\epsilon_{ij}(t)$. We prove this lemma, or equivalently $\epsilon(t) < 1/2(c-1)\beta_t$ for all $t \geq 1$, by induction. We first note that $\epsilon(1) < 1/4\beta_t$ is an assumption of the lemma. Next we show that $\epsilon(t+1) < 1/2(c-1)\beta_{t+1}$ if $\epsilon(t) < 1/2(c-1)\beta_t$.

By the fact that $|d_L - s_{ij}^*| \leq s_L^*$, $G_{ij}^c \subseteq N_{ij}^c$ and $s_L^* = 0$ for $L \in G_{ij}^c$, we obtain that

$$\begin{aligned}
\epsilon_{ij}(t+1)^2 = |s_{ij}^{(t)} - s_{ij}^*|^2 &= |\sqrt{\frac{\sum_{L\in N_{ij}^c} e^{-\beta_t s_L^{(t)}} d_L^2}{\sum_{L\in N_{ij}^c} e^{-\beta_t s_L^{(t)}}}} - s_{ij}^*|^2 \\
&\leq \frac{\sum_{L\in N_{ij}^c} e^{-\beta_t s_L^{(t)}} |d_L - s_{ij}^*|^2}{\sum_{L\in N_{ij}^c} e^{-\beta_t s_L^{(t)}}} \\
&\leq \frac{\sum_{L\in N_{ij}^c} e^{-\beta_t s_L^{(t)}} (s_L^*)^2}{\sum_{L\in N_{ij}^c} e^{-\beta_t s_L^{(t)}}} \\
&\leq \frac{\sum_{L\in B_{ij}^c} e^{-\beta_t s_L^{(t)}} (s_L^*)^2}{\sum_{L\in G_{ij}^c} e^{-\beta_t s_L^{(t)}}} \\
&\leq \frac{\sum_{L\in B_{ij}^c} e^{-\beta_t \sum_{e\in L}\epsilon_e(t)} (s_L^*)^2}{\sum_{L\in G_{ij}^c} e^{-\beta_t \sum_{e\in L}\epsilon_e(t)}} \\
&\leq \frac{1}{|G_{ij}^c|}\sum_{L\in B_{ij}^c} e^{2\beta_t(c-1)\epsilon(t)} e^{-\beta_t s_L^*}(s_L^*)^2.
\end{aligned} \tag{64}$$

By the induction assumption $\epsilon(t) < 1/2(c-1)\beta_t$ and then using the definition of $\lambda$, we have

$$\epsilon(t+1)^2 \leq \frac{e\sum_{L\in B_{ij}^c} e^{-\beta_t s_L^*}(s_L^*)^2}{|G_{ij}|} \leq \frac{e\lambda}{(1-\lambda)|B_{ij}|}\sum_{L\in B_{ij}^c} e^{-\beta_t s_L^*}(s_L^*)^2. \tag{65}$$

Combining the lemma assumptions and the definition of $M$ we have

$$\epsilon(t+1)^2 \leq \frac{e\lambda}{M(1-\lambda)\beta_t^2} = (\frac{r}{2(c-1)\beta_t})^2. \tag{66}$$

Therefore the lemma is proved by taking the square root of both sides. $\square$

*Proof of Lemma C.9.* To prove this lemma, we first prove an upper bound on the suprema of weakly dependent empirical processes. For an index set $\mathcal{A}$ and corresponding random variables $\{X_\alpha\}_{\alpha\in\mathcal{A}}$, we make the following definitions:

- A subset $\mathcal{A}'$ of $\mathcal{A}$ is independent if $\{X_\alpha\}_{\alpha\in\mathcal{A}'}$ is independent.
- A family of pairs $(\mathcal{A}_k, w_k)$ is a fractional cover of $\mathcal{A}$ if $\sum_k w_k 1_{\mathcal{A}_k} \geq 1_{\mathcal{A}}$.
- A fractional cover $(\mathcal{A}_k, w_k)$ is proper if each set $\mathcal{A}_k$ is independent.

*Lemma* D.4. Assume $\{X_\alpha\}_{\alpha\in I}$ are identically distributed according to $P$. Assume $\mathcal{F}$ is a countable set of functions that are all $P$-measurable and for all $f\in\mathcal{F}$, $\|f\|_\infty \leq 1$. Let $Z = \sup_{f\in\mathcal{F}}|\sum_{\alpha\in I} f(X_\alpha)|$. Assume $I$ admits a proper fractional cover $\{(I_j, w_j)\}_{j\in J}$, and $Z_j = \sup_{f\in\mathcal{F}}|\sum_{\alpha\in I_j} f(X_\alpha)|$. Let $\{p_j\}_{j\in J}$ be positive numbers such that $\sum_j p_j = 1$. Then

$$P(Z > \sum_j w_j \mathbb{E}Z_j + t) < \exp(-v\varphi(\frac{t}{Wv})) \tag{67}$$

where $v = 2\min_j \mathbb{E}Z_j + \sup_{f\in\mathcal{F}}\mathrm{Var}(f(X_\alpha))$ and $W = \sum_j w_j$.

*Proof.* We follow the proof strategy of [23]. By lemma 3.2 in [23] we can assume $(I_{\jmath},w_{\jmath})$ is an exact fractional cover of $I$. We have

$$Z = \sup_{f\in\mathcal{F}} |\sum_{\alpha\in I} f(X_\alpha)| \tag{68}$$

$$\leq \sup_{f\in\mathcal{F}} |\sum_{\alpha\in I}\sum_{\jmath} w_{\jmath} 1_{I_{\jmath}}(\alpha) f(X_\alpha)| \tag{69}$$

$$= \sup_{f\in\mathcal{F}} |\sum_{\jmath} w_{\jmath} \sum_{\alpha\in I} 1_{I_{\jmath}}(\alpha) f(X_\alpha)| \tag{70}$$

$$= \sup_{f\in\mathcal{F}} |\sum_{\jmath} w_{\jmath} \sum_{\alpha\in I_{\jmath}} f(X_\alpha)| \tag{71}$$

$$\leq \sum_{\jmath} w_{\jmath} \sup_{f\in\mathcal{F}} |\sum_{\alpha\in I_{\jmath}} f(X_\alpha)| = \sum_{\jmath} w_{\jmath} Z_{\jmath}. \tag{72}$$

Let $p_{\jmath}$ be any positive numbers such that $\sum_{\jmath} p_{\jmath} = 1$. By Jensen's inequality, for any $u > 0$,

$$\exp(u(Z - \sum_{\jmath}\mathbb{E}Z_{\jmath})) \leq \exp(\sum_{\jmath} p_{\jmath}\frac{uw_{\jmath}}{p_{\jmath}}(Z_{\jmath} - \mathbb{E}Z_{\jmath})) \leq \sum_{\jmath} p_{\jmath}\exp(\frac{uw_{\jmath}}{p_{\jmath}}(Z_{\jmath} - \mathbb{E}Z_{\jmath})). \tag{73}$$

Since $Z_{\jmath}$ is the supremum of a sum of independent random variables, by theorem 2.1 in [3] we have

$$\mathbb{E}\exp(\frac{uw_{\jmath}}{p_{\jmath}}(Z_{\jmath} - \mathbb{E}Z_{\jmath})) \leq \exp(\psi(-\frac{uw_{\jmath}}{p_{\jmath}})v_{\jmath}) \tag{74}$$

where $\psi(x) = e^{-x} - 1 + x$ and $v_{\jmath} = 2\mathbb{E}Z_{\jmath} + \sup_{f\in\mathcal{F}}\text{Var}(f(X_\alpha))$. Let $p_{\jmath} = w_{\jmath}/W$. By definition of $v$, $v = \min_{\jmath} v_{\jmath}$. By Markov's inequality we have

$$P(Z - \sum_{\jmath}\mathbb{E}Z_{\jmath} \geq t) \leq e^{-ut}\mathbb{E}e^{u(Z - \sum_{\jmath}\mathbb{E}Z_{\jmath})} \tag{75}$$

$$\leq e^{-ut}\frac{\sum_{\jmath} w_{\jmath} e^{\psi(-uW)v_{\jmath}}}{W} \tag{76}$$

$$\leq e^{-ut}\frac{\sum_{\jmath} w_{\jmath} e^{\psi(-uW)v}}{W} \tag{77}$$

$$= e^{-ut+\psi(-uW)v} \tag{78}$$

$$= e^{-ut+(e^{uW}-1-uW)v}. \tag{79}$$

Taking the minimum of the right hand side with respect to $u$ gives $P(Z \geq t) \leq e^{-v\varphi(t/Wv)}$. □

Now let's prove Lemma C.9. We slightly abuse the notation for simplicity. Throughout this proof we use $B_{i\jmath}$ as the set of all bad $i\jmath,c$-paths. To use Lemma D.4, we need to construct a proper fractional cover of $B_{i\jmath}^c$. Let $\Delta_1 = \lfloor |B_{i\jmath}^c|/cm_{c-1}\rfloor$. Note that by the regular E-R condition, we know that each $L \in B_{i\jmath}^c$ has at most $cm_{c-1}$ cycles that are correlated with $L$. By Hajnal-Szemerédi theorem, there exists a partition of $B_{i\jmath}^c$, namely $\{B_{i\jmath,k}^c\}_{k=1}^{cm_{c-1}}$, where for any $k$, $|B_{i\jmath,k}^c| = \Delta_1$ or $\Delta_1 + 1$, and all paths in $B_{i\jmath,k}^c$ are independent. This induces a proper fractional cover $(B_{i\jmath,k}^c, 1)$. By Lemma D.4, for any $t > 0$ we have

$$P(\sup_{f_r\in\mathcal{F}(\beta)}\sum_{L\in B_{i\jmath}^c} f_r(s_L^*) > t + cm_{c-1}\max_k \mathbb{E}Z_k) < \exp(-v\varphi(\frac{t}{cm_{c-1}v})). \tag{80}$$

where $v = 2\min_k \mathbb{E}Z_k + V(\beta)$.

By lemma 7 of [25] we know that $\mathbb{E}Z_k \leq C_1\sqrt{\log|B_{i\jmath,k}^c|/|B_{i\jmath,k}^c|}$. By $|B_{i\jmath,k}^c| \geq \Delta_1$ we know $\log|B_{i\jmath,k}^c|/|B_{i\jmath,k}^c| \leq \log\Delta_1/\Delta_1$.

By $\varphi(x) > \frac{x}{2}\ln(1+x)$ and the definition of $\Delta_1$, let $t = |B_{ij}^c|(2C_1\sqrt{\log\Delta_1/\Delta_1} + V(\beta))$ in (80), we have

$$\begin{aligned} &P\left(\sup_{f_\tau \in \mathcal{F}(\beta)} \frac{1}{|B_{ij}^c|}\sum_{L\in B_{ij}^c} f_\tau(s_L^*) > V(\beta) + (2C_1 + \frac{1}{\Delta_1})\sqrt{\frac{\log\Delta_1}{\Delta_1}}\right) \\ &< \exp\left(-\frac{\ln 2}{2}\Delta_1(2C_1\sqrt{\frac{\log\Delta_1}{\Delta_1}} + V(\beta))\right). \end{aligned} \tag{81}$$

By the definition of $m_{c-1}$ we know that $cm_{c-1} \sim \max(n^{c-3}p^{c-2}, n^\epsilon)$. Therefore $\Delta_1 = \Omega(\min(np, n^{c-2-\epsilon}p^{c-1}))$. Since $\Delta_1 \geq 1$, Lemma C.9 is proved by letting $K'' = 2C_1 + 1$. ☐

## E. Extension to any linear group with the metric induced by the Frobenius norm

Our algorithm LongSync can be extended to any linear group with the metric induced by the Frobenius norm. Let $\mathcal{D}_{\mathcal{G}}(\boldsymbol{G}_1, \boldsymbol{G}_2) = \|\boldsymbol{G}_1 - \boldsymbol{G}_2\|_F$ be such metric defined on a linear group $\mathcal{G}$. The update rule of LongSync becomes:

$$\begin{aligned} s_{ij}^{(t)} &= \Big(\sum_{L\in N_{ij}^c} w_L^{(t)} d_L^2 / z_{ij}^{(t)}\Big)^{1/2} \\ &= \Big(\sum_{L\in N_{ij}^c} w_L^{(t)} \mathcal{D}_{\mathcal{G}}^2(\boldsymbol{G}_L, \boldsymbol{G}_{ij}) / z_{ij}^{(t)}\Big)^{1/2} \\ &= \left(\Big(\sum_{L\in N_{ij}^c} w_L^{(t)} \|\boldsymbol{G}_L - \boldsymbol{G}_{ij}\|_F^2\Big) / z_{ij}^{(t)}\right)^{1/2} \\ &= \left(\Big(\Big\langle \sum_{L\in N_{ij}^c} \sqrt{w_L^{(t)}}\boldsymbol{G}_L, \sum_{L\in N_{ij}^c} \sqrt{w_L^{(t)}}\boldsymbol{G}_L\Big\rangle - 2\Big\langle \sum_{L\in N_{ij}^c} w_L^{(t)}\boldsymbol{G}_L, \boldsymbol{G}_{ij}\Big\rangle + \sum_{L\in N_{ij}^c} w_L^{(t)}\Big\langle \boldsymbol{G}_{ij}, \boldsymbol{G}_{ij}\Big\rangle\Big) / \sum_{L\in N_{ij}^c} w_L^{(t)}\right)^{1/2}. \end{aligned} \tag{82}$$

With the same $f_c$ and $g_c$ in 3.1, we have the following proposition:

**Proposition E.1.** *The update rule of of LongSync for any linear group in equation (82) is equivalent to the following matrix operations:*

$$\boldsymbol{S}^{(t)} = \left(\left(\Big\langle g_c(\sqrt{\boldsymbol{W}^{(t)}}, \boldsymbol{G}), g_c(\sqrt{\boldsymbol{W}^{(t)}}, \boldsymbol{G})\Big\rangle_{block} - 2\Big\langle g_c(\boldsymbol{W}^{(t)}, \boldsymbol{G}), \boldsymbol{G}\Big\rangle_{block}\right) \oslash f_c(\boldsymbol{W}^{(t)}) + \langle \boldsymbol{G}, \boldsymbol{G}\rangle_{block}\right)^{\odot 1/2} \tag{83}$$

*where* $\boldsymbol{W}^{(t+1)} = \boldsymbol{A} \odot \exp(-\beta_t \boldsymbol{S}^{(t)})$.

*Proof.* We prove the proposition by comparing the $ij$-th element of the right hand side of equation (83) with (82). By the definition of blockwise inner product, the $ij$-th block of the right hand side of equation (83) is

$$\left(\left(\Big\langle g_c(\sqrt{\boldsymbol{W}^{(t)}}, \boldsymbol{G})(i,j), g_c(\sqrt{\boldsymbol{W}^{(t)}}, \boldsymbol{G})(i,j)\Big\rangle - 2\Big\langle g_c(\boldsymbol{W}^{(t)}, \boldsymbol{G}), \boldsymbol{G}_{ij}\Big\rangle\right) / f_c(\boldsymbol{W}^{(t)}) + \langle \boldsymbol{G}_{ij}, \boldsymbol{G}_{ij}\rangle\right)^{1/2}.$$

Note that by definition of $g_c$, $g_c(\sqrt{\boldsymbol{W}(t)}, \boldsymbol{G})(i,j) = \sum_{L\in N_{ij}^c} \sqrt{w_L^{(t)}}\boldsymbol{G}_L$, and $g_c(\boldsymbol{W}(t), \boldsymbol{G})(i,j) = \sum_{L\in N_{ij}^c} w_L^{(t)}\boldsymbol{G}_L$. By the definition of $f_c$, $f_c(\boldsymbol{W}^{(t)})(i,j) = \sum_{L\in N_{ij}^c} w_L^{(t)}$. By directly comparing the terms we know that the right hand side of equation (83) is the same as (82). ☐

In view of this vectorized update rule, we propose the vectorized LongSync iterations for any linear group with $l_2$ metric in algorithm 2.

We remark that the theory of LongSync can also be adapted as long as the group is 'well-conditioned', i.e. there exists constants $M_{\mathcal{G}}$ and $m_{\mathcal{G}}$ only depending on $\mathcal{G}$ such that for any $\boldsymbol{G} \in \mathcal{G}$, the absolute value of the eigenvalues of $\boldsymbol{G}$ is between $m_{\mathcal{G}}$ and $M_{\mathcal{G}}$.

**Algorithm 2** (LongSync for any linear group)

**Input:** pairwise measurement matrix $\boldsymbol{G}$, adjacency matrix $\boldsymbol{A}\in[0,1]^{n\times n}$, cycle length $c$, positive parameters $\{\beta_t\}_{t\geq 1}$, time step $T$

$\boldsymbol{W}^{(0)}(i,j)\leftarrow\boldsymbol{A}$

**for** $t=0:T$ **do**

$$\boldsymbol{S}^{(t)}\leftarrow\left(\left(\left\langle g_c(\sqrt{\boldsymbol{W}^{(t)}},\boldsymbol{G}),g_c(\sqrt{\boldsymbol{W}^{(t)}},\boldsymbol{G})\right\rangle_{\text{block}}-2\left\langle g_c(\boldsymbol{W}^{(t)},\boldsymbol{G}),\boldsymbol{G}\right\rangle_{\text{block}}\right)\oslash f_c(\boldsymbol{W}^{(t)})+\langle\boldsymbol{G},\boldsymbol{G}\rangle_{\text{block}}\right)^{\odot 1/2}$$

$$\boldsymbol{W}^{(t+1)}\leftarrow\boldsymbol{A}\odot\exp(-\beta_t\boldsymbol{S}^{(t)})$$

**end for**

**Output:** edge weights $\boldsymbol{W}^{(T+1)}$, corruption levels $\boldsymbol{S}^{(T)}$