Probabilistic Permutation Synchronization using the Riemannian Structure of the Birkhoff Polytope

Tolga Birdal [1] Umut Şimşekli [2]
[1] Fakultät für Informatik, Technische Universität München, 85748 München, Germany
[2] LTCI, Télécom ParisTech, Université Paris-Saclay, 75013 Paris, France

Abstract

We present an entirely new geometric and probabilistic approach to synchronization of correspondences across multiple sets of objects or images. In particular, we present two algorithms: (1) Birkhoff-Riemannian L-BFGS for optimizing the relaxed version of the combinatorially intractable cycle consistency loss in a principled manner, (2) Birkhoff-Riemannian Langevin Monte Carlo for generating samples on the Birkhoff Polytope and estimating the confidence of the found solutions. To this end, we first introduce the very recently developed Riemannian geometry of the Birkhoff Polytope. Next, we introduce a new probabilistic synchronization model in the form of a Markov Random Field (MRF). Finally, based on the first order retraction operators, we formulate our problem as simulating a stochastic differential equation and devise new integrators. We show on both synthetic and real datasets that we achieve high quality multi-graph matching results with faster convergence and reliable confidence/uncertainty estimates.

1. Introduction

Correspondences fuel a large variety of computer vision applications such as structure-from-motion (SfM) [73], SLAM [61], 3D reconstruction [24, 10, 8], camera relocalization [71], image retrieval [52] and 3D scan stitching [45, 26]. In a typical scenario, given two scenes, an initial set of 2D/3D keypoints is first identified. Then the neighborhood of each keypoint is summarized with a local descriptor [55, 27] and keypoints in the given scenes are matched by associating the mutually closest descriptors. In a majority of practical applications, multiple images or 3D shapes are under consideration and ascertaining such two-view or *pairwise* correspondences is simply not sufficient. This necessitates a further refinement ensuring global consistency. Unfortunately, at this stage even the well developed pipelines acquiesce either heuristic/greedy refinement [25] or incorporate costly geometric cues related to the linking of individual correspondence estimates into a globally coherent whole [35, 73, 84].

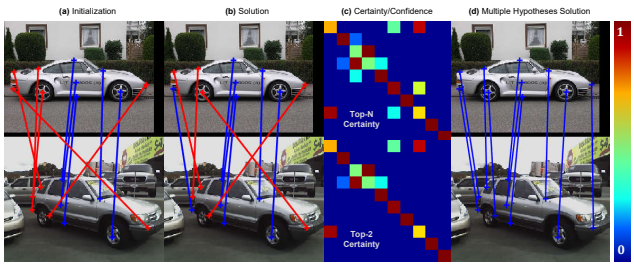


Figure 1. Our algorithm robustly solves the multiway image matching problem (**a**, **b**) and provides confidence maps (**c**) that can be of great help in further improving the estimates (**d**). The bar on the right is used to assign colors to confidences. For the rest, incorrect matches are marked in red and correct ones in blue.

In this paper, by using the fact that correspondences are *cycle consistent* [1], we propose two novel algorithms for refining the assignments across multiple images/scans (nodes) in a *multi-way graph* and for estimating assignment confidences, respectively. We model the correspondences between image pairs as *relative*, *total* permutation matrices and seek to find *absolute* permutations that re-arrange the detected keypoints to a single canonical, global order. This problem is known as *map* or *permutation synchronization* [64, 81]. Even though in many practical scenarios matches are only partially available, when shapes are complete and the density of matches increases, total permutations can suffice [42].

Similar to many well received works [97, 72], we relax the sought permutations to the set of doubly-stochastic (DS) matrices. We then consider the geometric structure of DS, the *Birkhoff Polytope* [11]. We are - to the best of our knowledge, for the first time introducing and applying the recently developed Riemannian geometry of the Birkhoff Polytope [30] to tackle challenging problems of computer vision. Note that lack of this geometric understanding caused plenty of obstacles for scholars dealing with our problem [72, 88]. By the virtue of a first order retraction, we can use the recent Riemannian limited-memory

[1] Composition of correspondences for any circular path arrives back at the start node.

BFGS (LR-BFGS) algorithm [95] to perform a maximum-a-posteriori (MAP) estimation of the parameters of the consistency loss. We coin our variation as *Birkhoff-LRBFGS*.

At the next stage, we take on the challenge of confidence/uncertainty estimation for the problem at hand by drawing samples on the Birkhoff Polytope and estimating the empirical posterior distribution. To achieve this, we first formulate a new geodesic stochastic differential equation (SDE). Our SDE is based upon the Riemannian Langevin Monte Carlo (RLMC) [36, 91, 66] that is efficient and effective in sampling from Riemannian manifolds with *true* exponential maps. Note that similar stochastic gradient geodesic MCMC (SG-MCMC) [54, 15] tools have already been used in the context of synchronization of spatial rigid transformations whose parameters admit an analytically defined geodesic flow [9]. Unfortunately, for our manifold the retraction map is only up to first order and hence we cannot use off-the-shelf schemes. Alleviating this nuisance, we further contribute a novel numerical integrator to solve our SDE by replacing the intractable exponential map of DS matrices by the approximate retraction map. This leads to another new algorithm: *Birkhoff-RLMC*.

In a nutshell, our contributions are:

1. We function as an ambassador and introduce the Riemannian geometry of the Birkhoff Polytope [30] to solve problems in computer vision.
2. We propose a new probabilistic model for the permutation synchronization problem.
3. We minimize the cycle consistency loss via a Riemannian-LBFGS algorithm and outperfom the state-of-the-art both in recall and in runtime.
4. Based upon the Langevin mechanics, we introduce a new SDE and a numerical integrator to draw samples on the high dimensional and complex manifolds with approximate retractions, such as the Birkhoff Polytope. This lets us estimate the confidence maps, which can aid in improving the solutions and spotting consistency violations or outliers.

Note that the tools developed herewith can easily extend beyond our application and would hopefully facilitate promising research directions regarding the combinatorial optimization problems in computer vision.

2. Related Work

Permutation synchronization is an emerging domain of study due to its wide applicability, especially for the problems in computer vision. We now review the developments in this field, as chronologically as possible. Note that multi-way graph matching problem formulations involving spatial geometry are well studied [22, 58, 51, 33, 92, 23], as well as transformation synchronization [87, 17, 83, 86, 5, 6, 38]. For brevity, we omit these literature and focus on works that explicitly operate on correspondence matrices.

The first applications of *synchronization*, a term coined by Singer [78, 77], to correspondences only date back to early 2010s [62]. Pachauri *et al*. [64] gave a formal definition and devised a spectral technique. The same authors quickly extended their work to Permutation Diffusion Maps [63] finding correspondence between images. Unfortunately, this first method was quadratic in the number of images and hence was not computationally friendly. In a sequel of works called *MatchLift*, Huang, Chen and Guibas [42, 19] were the firsts to cast the problem of estimating cycle-consistent maps as finding the closest positive semidefinite matrix to an input matrix. They also addressed the case of partial permutations. Due to the semidefinite programming (SDP) involved, this perspective suffered from high computational cost in real applications. Similar to Pachauri [64], for N images and M edges, this method required computing an eigendecomposition of an $NM \times NM$ matrix. Zhou *et al*. [98] then introduced *MatchALS*, a new low-rank formulation with nuclear-norm relaxation, globally solving the joint matching of a set of images without the need of SDP. Yu *et al*. [94] formulated a synchronization energy similar to our method and proposed proximal Gauss-Seidel methods for solving a relaxed problem. However, unlike us, this paper did not use the geometry of the constraints or variables and thereby resorted to complicated optimization procedures involving Frank-Wolfe subproblems for global constraint satisfaction. Arrigoni *et al*. [4] and Maset *et al*. [4] extended Pachauri [64] to operate on partial permutations using spectral decomposition. To do so, they considered the symmetric inverse semigroup of the partial matches that are typically hard to handle. Their closed form methods did not need initialization steps to synchronize, but also did not establish an explicit cycle consistency. Tang *et al*. [82] opted to use ordering heuristics improving upon Pachauri [64]. Cosmo *et al*. [23] brought an interesting solution to the problem of estimating consistent correspondences between multiple 3D shapes, without requiring initial pairwise solutions as input. Schiavinato and Torsello [72] tried to overcome the lack of group structure of the Birkhoff polytope by transforming any graph-matching problem into a multi-graph matching one. Bernard *et al*. [7] used an NMF-based approach to generate a cycle-consistent synchronization. Park and Yoon [65] used multi-layer random walks framework to address the global correspondence search problem of multi-attributed graphs. Starting from a multi-layer random-walks initialization, the authors proposed a robust solver by iterative reweighting. Hu *et al*. [41] revisited the *MatchLift* and developed a scalable, distributed solution with the help of ADMMs, called *DMatch*. Their idea of splitting the input into sub-collections can still lead to global consistency under mild conditions while improving the efficiency. Finally, Wang *et al*. [88] made use of the domain knowledge and

added the geometric consistency of image coordinates as a low-rank term to increase the recall.

The aforementioned works have neither considered the Riemennian structure of the common Birkhoff convex relaxation nor have they provided a probabilistic framework, which can pave the way to uncertainty estimation while simultaneously solving the optimization problem. This is what we propose in this work.

3. Preliminaries and Technical Background

Definition 1 (Permutation Matrix)**.** *A permutation matrix is defined as a sparse, square binary matrix, where each column and each row contains only a single true (1) value:*

$$\mathcal{P}_n := \{\mathbf{P} \in \{0,1\}^{n\times n} : \mathbf{P}\mathbf{1}_n = \mathbf{1}_n \,,\, \mathbf{1}_n^\top \mathbf{P} = \mathbf{1}_n^\top\}. \quad (1)$$

where $\mathbf{1}_n$ denotes a n-dimensional ones vector. Every $\mathbf{P} \in \mathcal{P}^n$ is a total permutation matrix and $P_{ij} = 1$ implies that element i is mapped to element j. Permutation matrices are the only strictly non-negative elements of the orthogonal group $\mathcal{P}_n \in \mathcal{O}_n = \{\mathbf{O} : \mathbf{O}^\top\mathbf{O} = \mathbf{I}\}$, a special case of the Stiefel manifold of m—frames in $\mathbb{R}_n$ when $m = n$.

Definition 2 (Center of Mass)**.** *The center of mass for all the permutations on n objects is defined in $\mathbb{R}^{n\times n}$ as [67]:*

$$\mathbf{C}_n = \frac{1}{n!}\sum\nolimits_{\mathbf{P}_i\in\mathcal{P}_n} \mathbf{P}_i = \frac{1}{n!}(n-1)!\mathbf{1}_n\mathbf{1}_n^\top = \frac{1}{n}\mathbf{1}_n\mathbf{1}_n^\top. \quad (2)$$

Notice that $\mathbf{C}_n \notin \mathcal{P}^n$ as shown in Fig. 2.

Definition 3 (Relative Permutation)**.** *We define a permutation matrix to be **relative** if it is the ratio (or difference) of two group elements $(i \to j)$: $\mathbf{P}_{ij} = \mathbf{P}_i\mathbf{P}_j^\top$.*

Definition 4 (Permutation Synchronization Problem)**.** *Given a redundant set of measures of ratios $\{\mathbf{P}_{ij}\} : (i,j) \in \mathcal{E} \subset \{1,\dots,N\} \times \{1,\dots,N\}$, where $\mathcal{E}$ denotes the set of the edges of a directed graph of N nodes, the permutation synchronization [64] can be formulated as the problem of recovering $\{\mathbf{P}_i\}$ for $i = 1,\dots,N$ such that the group consistency constraint is satisfied: $\mathbf{P}_{ij} = \mathbf{P}_i\mathbf{P}_j^{-1}$.*

If the input data is noise-corrupted, this *consistency* will not hold and to recover the *absolute permutations* $\{\mathbf{P}_i\}$, some form of a *consistency error* is minimized. Typically, any form of minimization on the discrete space of permutations is intractable and these matrices are relaxed by their *doubly-stochastic* counterparts [14, 97, 72] (see Fig. 2).

Definition 5 (Doubly Stochastic (DS) Matrix)**.** *A DS matrix is a non-negative, square matrix whose rows and columns sum to 1. The set of DS matrices is defined as:*

$$\mathcal{DP}_n = \{\ \mathbf{X} \in \mathbb{R}_+^{n\times n} : \sum_{i=1}^{n} x_{ij} = 1 \wedge \sum_{j=1}^{n} x_{ij} = 1\ \}. \quad (3)$$

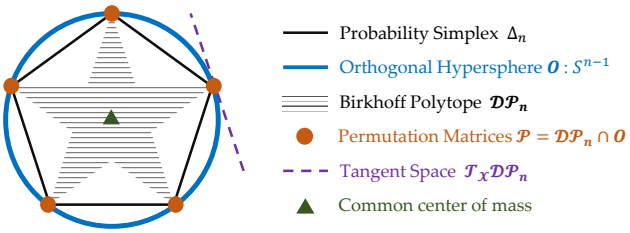


Figure 2. Simplified (matrices are vectorized) illustration of geometries we consider: (i) Δ_n is convex, (ii) $\mathcal{DP}_n$ is strictly contained in Δ_n. In low dimensions, such configuration cannot exist as there is no convex shape that touches Δ_n only on the corners.

Theorem 1 (Birkhoff-von Neumann Theorem)**.** *The convex hull of the set of all permutation matrices is the set of doubly-stochastic matrices and there exists a potentially non-unique $\boldsymbol{\theta}$ such that any DS matrix can be expressed as a linear combination of k permutation matrices [11, 46]:*

$$\mathbf{X} = \theta_1\mathbf{P}_1 + \cdots + \theta_k\mathbf{P}_k \,,\, \theta_i > 0 \wedge \boldsymbol{\theta}^\top\mathbf{1}_k = 1. \quad (4)$$

While finding the minimum k is shown to be NP-hard [31], by Marcus-Ree theorem, we know that there exists one constructible decomposition where $k < (n-1)^2 + 1$.

Definition 6 (Birkhoff Polytope)**.** *The multinomial manifold of DS matrices is incident to the convex object called the Birkhoff Polytope [11], an $(n-1)^2$ dimensional convex submanifold of the ambient $\mathbb{R}^{n\times n}$ with $n!$ vertices: $\mathcal{B}_n \equiv \mathcal{DP}_n$. We use $\mathcal{DP}_n$ to refer to the Birkhoff Polytope.*

It is interesting to see that this convex polytope is co-centered with $\mathcal{P}_n$ at $\mathbf{C}_n$, $\mathbf{C}_n \in \mathcal{DP}_n$ and over-parameterizes the convex hull of the permutation vectors, the *permutahedron* [37]. $\mathcal{P}_n$ can now be considered as an orthogonal subset of $\mathcal{DP}_n$: $\mathcal{P}_n = \{\mathbf{X} \in \mathcal{DP}_n : \mathbf{X}\mathbf{X}^\top = \mathbf{I}\}$, i.e. the discrete set of permutation matrices is the intersection of the convex set of DS matrices and the $\mathcal{O}_n$.

3.1. Riemannian Geometry of the Birkhoff Polytope

Recently, Douik *et al.* [30] endowed $\mathcal{DP}_n$ with the Fisher information metric, resulting in the Riemannian manifold of $\mathcal{DP}_n$. To the best of our knowledge, we are the first to exploit this manifold in the domain of computer vision, and hence will now recall the main results of Douik *et al.* [30] and summarize the main constructs of Riemannian optimization on $\mathcal{DP}_n$. The proofs can be found in [30].

Definition 7 (Tangent Space and Bundle)**.** *The tangent bundle is referred to as the union of all tangent spaces $\mathcal{TDP}_n = \cup_{\mathbf{X}\in\mathcal{DP}_n}\mathcal{T}_\mathbf{X}\mathcal{DP}_n$ one of which is defined as:*

$$\mathcal{T}_\mathbf{X}\mathcal{DP}_n := \{\mathbf{Z} \in \mathbb{R}^{n\times n} : \mathbf{Z}\mathbf{1}_n = \mathbf{0}_n \,,\, \mathbf{Z}^\top\mathbf{1}_n = \mathbf{0}_n\}. \quad (5)$$

Theorem 2. *The projection operator $\Pi_\mathbf{X}(\mathbf{Y}), \mathbf{Y} \in \mathcal{DP}_n$ onto the tangent space of $\mathbf{X} \in \mathcal{DP}_n$, $\mathcal{T}_\mathbf{X}\mathcal{DP}_n$ is written as:*

$$\Pi_\mathbf{X}(\mathbf{Y}) = \mathbf{Y} - (\boldsymbol{\alpha}\mathbf{1}^\top + \mathbf{1}\boldsymbol{\beta}^\top) \odot \mathbf{X}, \quad \text{with} \quad (6)$$

$$\boldsymbol{\alpha} = (\mathbf{I} - \mathbf{X}\mathbf{X}^\top)^+(\mathbf{Y} - \mathbf{X}\mathbf{Y}^\top)\mathbf{1}, \quad \boldsymbol{\beta} = \mathbf{Y}^\top\mathbf{1} - \mathbf{X}^\top\boldsymbol{\alpha},$$

$^+$ depicts the left pseudo-inverse and $\odot$ the Hadamard product. Note that there exists a numerically more stable way to compute the same concise formulation of $\Pi_{\mathbf{X}}(\mathbf{Y})$ [30].

Theorem 3. *For a vector $\boldsymbol{\xi}_{\mathbf{X}} \in \mathcal{T}_{\mathbf{X}}\mathcal{DP}_n$ lying on the tangent space of $\mathbf{X} \in \mathcal{DP}_n$, the first order retraction map $R_{\mathbf{X}}$ is given as follows:*

$$R_{\mathbf{X}}(\boldsymbol{\xi}_{\mathbf{X}}) = \Pi(\mathbf{X} \odot \exp(\boldsymbol{\xi}_{\mathbf{X}} \oslash \mathbf{X})), \tag{7}$$

where the operator Π denotes the projection onto $\mathcal{DP}_n$, efficiently computed using the Sinkhorn algorithm [79] and $\oslash$ is the Hadamard division.

Plis *et al*. [67] showed that on the n-dimensional Birkhoff Polytope all permutations are equidistant from the center of mass $\mathbf{C}_n$, and thus the extreme points of $\mathcal{DP}_n$, that are the permutation matrices, are located on an $(n-1)^2$-dimensional hypersphere $S^{(n-1)^2}$ of radius $\sqrt{n-1}$, centered at $\mathbf{C}_n$. This hypersphere is incident to the Birkhoff Polytope on the vertices.

Proposition 1. *The gap as a ratio between $\mathcal{DP}_n$ and both $\mathcal{S}^{(n-1)^2}$ and $\mathcal{O}_n$ grows to infinity as n grows.*

The proof is given in the supplementary document. While there exists polynomial time projections of the $n!$-element permutation space onto the continuous hypersphere representation and back [67], Prop. 1 prevents us from using hypersphere relaxations, as done in preceding works [67, 96].

4. Proposed Probabilistic Model

We assume that we are provided a set of pairwise, *total* permutations $\mathbf{P}_{ij} \in \mathcal{P}_n$ for $(i,j) \in \mathcal{E}$ and we are interested in finding the underlying *absolute permutations* $\mathbf{X}_i$ for $i \in \{1,\dots,N\}$ with respect to a common origin (e.g. $\mathbf{X}_1 = \mathbf{I}$, the identity matrix). We seek absolute permutations that would respect the consistency of the underlying graph structure. For conciseness, we also restrict our setting to total permutations, and leave the extension to partial permutations, which live on the *monoid*, as a future study. Because operating directly on $\mathcal{P}_n$ would require us to solve a combinatorial optimization problem and because of the lack of a manifold structure for $\mathcal{P}_n$, we follow the popular approach [53, 93, 56] and relax the domain of the absolute permutations by assuming that each $\mathbf{X}_i \in \mathcal{DP}_n$.

We formulate the permutation synchronization problem in a probabilistic context where we treat the pairwise relative permutations as *observed* random variables and the absolute ones as *latent* random variables. In particular, our probabilistic construction enables us to cast the synchronization problem as inferential in the model. With a slight abuse of notation, in the rest of the paper, we will denote $\mathbf{P} \equiv \{\mathbf{P}_{ij}\}_{(i,j)\in\mathcal{E}}$ and $\mathbf{X} \equiv \{\mathbf{X}_i\}_{i=1}^N$, all the observations and all the latent variables, respectively.

A typical way to build a probabilistic model is to first choose the prior distributions on $\mathcal{DP}_n$ for each $\mathbf{X}_i$ and then choose a conditional distribution on $\mathcal{P}_n$ for each $\mathbf{X}_{ij}$ given the latent variables. Unfortunately, standard parametric distributions neither exist on $\mathcal{DP}_n$ nor on $\mathcal{P}_n$. The variational *stick breaking* [53] yields an *implicitly* defined PDF on $\mathcal{DP}_n$ and is not able to provide direct control on the resulting distribution. Defining Kantorovich distance-based distributions over the permutation matrices is possible [21], yet these models incur high computational costs since they would require solving optimal transport problems during inference. For these reasons, instead of constructing a hierarchical probabilistic model, we will directly model the full joint distribution of $\mathbf{P}$ and $\mathbf{X}$.

We propose a probabilistic model where we assume the full joint distribution admits the following factorized form:

$$p(\mathbf{P}, \mathbf{X}) = \frac{1}{Z}\prod\nolimits_{(i,j)\in\mathcal{E}} \psi(\mathbf{P}_{ij}, \mathbf{X}_i, \mathbf{X}_j), \tag{8}$$

where Z denotes the normalization constant with

$$Z := \sum_{\mathbf{P}\in\mathcal{P}_n^{|\mathcal{E}|}} \int_{\mathcal{DP}_n^N} \prod_{(i,j)\in\mathcal{E}} \psi(\mathbf{P}_{ij}, \mathbf{X}_i, \mathbf{X}_j)\, \mathrm{d}\mathbf{X}, \tag{9}$$

and ψ is called the 'clique potential' that is defined as:

$$\psi(\mathbf{P}_{ij}, \mathbf{X}_i, \mathbf{X}_j) \triangleq \exp(-\beta\|\mathbf{P}_{ij} - \mathbf{X}_i\mathbf{X}_j^\top\|_{\mathrm{F}}^2). \tag{10}$$

Here $\|\cdot\|_{\mathrm{F}}$ denotes the Frobenius norm, $\beta \in \mathbb{R}_+$ is the *dispersion* parameter that controls the spread of the distribution. Note that the model is a Markov random field [48].

Let us take a closer look at the proposed model. If we define $\mathbf{X}_{ij} := \mathbf{X}_i\mathbf{X}_j^\top \in \mathcal{DP}_n$, then by Thm. 1, we have the following decomposition for each $\mathbf{X}_{ij}$:

$$\mathbf{X}_{ij} = \sum\nolimits_{b=1}^{B_{ij}} \theta_{ij,b}\mathbf{M}_{ij,b}, \quad \sum\nolimits_{b=1}^{B_{ij}} \theta_{ij,b} = 1, \tag{11}$$

where B_{ij} is a positive integer, each $\theta_{ij,b} \geq 0$, and $\mathbf{M}_{ij,b} \in \mathcal{P}_n$. The next result states that we have an equivalent hierarchical interpretation for the proposed model:

Proposition 2. *The probabilistic model defined in Eq. 8 implies the following hierarchical decomposition:*

$$p(\mathbf{X}) = \frac{1}{C}\exp\Big(-\beta\sum_{(i,j)\in\mathcal{E}}\|\mathbf{X}_{ij}\|^2\Big)\prod_{(i,j)\in\mathcal{E}} Z_{ij} \tag{12}$$

$$p(\mathbf{P}_{ij}|\mathbf{X}_i, \mathbf{X}_j) = \frac{1}{Z_{ij}}\exp\Big(2\beta\,\mathrm{tr}(\mathbf{P}_{ij}^\top\mathbf{X}_{ij})\Big) \tag{13}$$

where C and Z_{ij} are normalization constants. Besides, for all i, j, $Z_{ij} \geq \prod_{b=1}^{B_{ij}} f(\beta, \theta_{ij,b})$, where f is a positive function that is increasing in both β and $\theta_{ij,b}$.

The proof is given in the supplementary and is based on the simple decomposition $p(\mathbf{P}, \mathbf{X}) = p(\mathbf{X})p(\mathbf{P}|\mathbf{X})$. This hierarchical point of view lets us observe some interesting properties: (**1**) the likelihood $p(\mathbf{P}_{ij}|\mathbf{X}_i, \mathbf{X}_j)$ mainly depends on the term $\mathrm{tr}(\mathbf{P}_{ij}^\top \mathbf{X}_{ij})$ that measures the data fitness. We aptly call this term the 'soft Hamming distance' between $\mathbf{P}_{ij}$ and $\mathbf{X}_{ij}$ since it would correspond to the actual Hamming distance between two permutations if $\mathbf{X}_i, \mathbf{X}_j$ were permutation matrices [49]. (**2**) On the other hand, the prior distribution contains two competing terms: (i) the term Z_{ij} favors large $\theta_{ij,b}$, which would push $\mathbf{X}_{ij}$ towards the corners of the Birkhoff polytope, (ii) the term $\|\mathbf{X}_{ij}\|_{\mathrm{F}}^2$ acts as a regularizer on the latent variables and attracts them towards the center of the Birkhoff polytope $\mathbf{C}_n$ (cf. Dfn. 2), which will be numerically beneficial for the inference algorithms that will be developed in the following section.

5. Inference Algorithms

We can now formulate the permutation synchronization problem as a probabilistic inference problem, where we will be interested in the following quantities:

1. Maximum a-posteriori (MAP):
$$\mathbf{X}^\star = \operatorname*{arg\,max}_{\mathbf{X} \in \mathcal{DP}_n^N} \log p(\mathbf{X}|\mathbf{P}) \tag{14}$$
where $\log p(\mathbf{X}|\mathbf{P}) =^+ -\beta \sum_{(i,j)\in\mathcal{E}} \|\mathbf{P}_{ij} - \mathbf{X}_i \mathbf{X}_j^\top\|_{\mathrm{F}}^2$, and $=^+$ denotes equality up to an additive constant.
2. The full posterior distribution: $p(\mathbf{X}|\mathbf{P}) \propto p(\mathbf{P}, \mathbf{X})$.

The MAP estimate is often easier to obtain and useful in practice. On the other hand, characterizing the full posterior can provide important additional information, such as *uncertainty*; however, not surprisingly it is a much harder task. In addition to the usual difficulties associated with these tasks, in our context we are facing extra challenges due to the non-standard manifold of our latent variables.

5.1. Maximum A-Posteriori Estimation

The MAP estimation problem can be cast as a minimization problem on $\mathcal{DP}_n$, given as follows:

$$\mathbf{X}^\star = \operatorname*{arg\,min}_{\mathbf{X} \in \mathcal{DP}_n^N} \Big\{ U(\mathbf{X}) := \sum\nolimits_{(i,j)\in\mathcal{E}} \|\mathbf{P}_{ij} - \mathbf{X}_i \mathbf{X}_j^\top\|_{\mathrm{F}}^2 \Big\}$$

where U is called the potential energy function. We observe that the choice of the dispersion parameter has no effect on the MAP estimate. Although this optimization problem resembles conventional norm minimization, the fact that $\mathbf{X}$ lives in the cartesian product of Birkhoff polytopes renders the problem very complicated.

Thanks to the retraction operator over the Birkhoff polytope (cf. Thm. 3), we are able to use several Riemannian optimization algorithms [80], without resorting to projection-based updates. In this study, we use the recently proposed Riemannian limited-memory BFGS (LR-BFGS) [44], a powerful optimization technique that attains faster convergence rates by incorporating local geometric information in an efficient manner. This additional piece of information is obtained through an approximation of the inverse Hessian, which is computed on the most recent values of the past iterates with linear time- and space-complexity in the dimension of the problem. We give more detail on LR-BFGS in our supp. material. The detailed description of the algorithm can be found in [44, 95].

Finally, we round the resulting approximate solutions into a feasible one via Hungarian algorithm [60], obtaining binary permutation matrices.

5.2. Posterior Sampling via Riemannian Langevin Monte Carlo with Retractions

In this section we will develop a Markov Chain Monte Carlo (MCMC) algorithm for generating samples from the posterior distribution $p(\mathbf{X}|\mathbf{P})$, by borrowing ideas from [75, 54, 9]. Once such samples are generated, we will be able to quantify the uncertainty in our estimation by using the generated samples.

The dimension and complexity of the Birkhoff manifold makes it very challenging to generate samples on $\mathcal{DP}_n$ or its product spaces and to the best of our knowledge there is no Riemannian MCMC algorithm that is capable of achieving this. There are existing Riemannian MCMC algorithms [15, 54], which are able to draw samples on embedded manifolds; however, they require the exact exponential map to be analytically available, which in our case, can only be approximated by the retraction map at best.

To this end, we develop an algorithmically simpler yet effective algorithm. Let the posterior density of interest be $\pi_{\mathcal{H}}(\mathbf{X}) := p(\mathbf{X}|\mathbf{P}) \propto \exp(-\beta U(\mathbf{X}))$ with respect to the Hausdorff measure. We then define an *embedding* $\xi : \mathbb{R}^{N(n-1)^2} \mapsto \mathcal{DP}_n^N$ such that $\xi(\tilde{\mathbf{X}}) = \mathbf{X}$ for $\tilde{\mathbf{X}} \in \mathbb{R}^{N(n-1)^2}$. By the area formula (cf. Thm. 1 in [29]), we have the following expression for the embedded posterior density π_λ (with respect to the Lebesgue measure):

$$\pi_{\mathcal{H}}(\mathbf{x}) = \pi_\lambda(\tilde{\mathbf{X}}) / \sqrt{|\mathbf{G}(\tilde{\mathbf{X}})|}, \tag{15}$$

where $\mathbf{G}$ denotes the Riemann metric tensor.

We then consider the following stochastic differential equation (SDE), which is a slight modification of the SDE that is used to develop the Riemannian Langevin Monte Carlo algorithm [36, 91, 66]:

$$\mathrm{d}\tilde{\mathbf{X}}_t = (-\mathbf{G}^{-1} \nabla_{\tilde{\mathbf{X}}} U_\lambda(\tilde{\mathbf{X}}_t) + \mathbf{\Gamma}_t)\mathrm{d}t + \sqrt{2/\beta \mathbf{G}^{-1}}\mathrm{dB}_t,$$

where B_t denotes the standard Brownian motion and $\mathbf{\Gamma}_t$ is called the correction term that is defined as follows: $[\mathbf{\Gamma}_t(\tilde{\mathbf{X}})]_i = \sum_{j=1}^{N(n-1)^2} \partial [\boldsymbol{G}_t^{-1}(\tilde{\mathbf{X}})]_{ij} / \partial \tilde{\mathbf{X}}_j$.

Figure 3. Sample images and manually annotated correspondences from the challenging Willow dataset [20]. Images are plotted in pairs (there are multiple) and in gray for better viewing.

By Thm. 1 of [57], it is easy to show that the solution process $(\tilde{\mathbf{X}}_t)_{t\geq 0}$ leaves the embedded posterior distribution π_λ invariant. Informally, this result means that if we could exactly simulate the continuous-time process $(\tilde{\mathbf{X}}_t)_{t\geq 0}$, the distribution of the sample paths would converge to the embedded posterior distribution π_λ, and therefore the distribution of $\xi(\tilde{\mathbf{X}}_t)$ would converge to $\pi_{\mathcal{H}}(\mathbf{X})$. However, unfortunately it is not possible to exactly simulate these paths and therefore we need to consult approximate algorithms.

A possible way to numerically simulate the SDE would be to use standard discretization tools, such as the Euler-Maruyama integrator [18]. However, this would require knowing the analytical expression of ξ and constructing $\mathbf{G}_t$ and $\mathbf{\Gamma}_t$ at each iteration. On the other hand, recent results have shown that we can simulate SDEs directly on their original manifolds by using geodesic integrators [15, 54, 39], which bypasses these issues altogether. Yet, these approaches require the exact exponential map of the manifold to be analytically available, restricting their applicability in our context.

Inspired by the recent manifold optimization algorithms [85], we propose to replace the exact, intractable exponential map arising in the geodesic integrator with the tractable retraction operator given in Thm. 3. We develop our recursive scheme, we coin as *retraction Euler integrator*:

$$\mathbf{V}_i^{(k+1)} = \Pi_{\mathbf{X}_i^{(k)}}(h\nabla_{\mathbf{X}_i} U(\mathbf{X}_i^{(k)}) + \sqrt{2h/\beta}\mathbf{Z}_i^{(k+1)}) \quad (16)$$

$$\mathbf{X}_i^{(k+1)} = R_{\mathbf{X}_i^{(k)}}(\mathbf{V}_i^{(k+1)}), \qquad \forall i \in \{1,\dots,N\} \quad (17)$$

where $h > 0$ denotes the step-size, k denotes the iterations, $\mathbf{Z}_i^{(k)}$ denotes standard Gaussian random variables in $\mathbb{R}^{n\times n}$, $\mathbf{X}_i^{(0)}$ denotes the initial absolute permutations. The derivation of this scheme is similar to [54] and we provide more detailed information in the supplementary material. To the best of our knowledge, the convergence properties of the geodesic integrator that is approximated with a retraction operator have not yet been analyzed. We leave this analysis as a futurework, which is beyond the scope of this study.

We note that the term $\|\mathbf{X}_{ij}\|_{\mathrm{F}}^2$ plays an important role in the overall algorithm since it prevents the latent variables $\mathbf{X}_i$ to go the extreme points of the Birkhoff polytope, where the retraction operator becomes inaccurate. We also note that, when $\beta \to \infty$, the distribution $\pi_{\mathcal{H}}$ concentrates on the global optimum $\mathbf{X}^\star$ and the proposed retraction Euler integrator becomes the Riemannian gradient descent with a retraction operator.

6. Experiments and Evaluations

6.1. Real Data

2D Multi-image Matching We run our method to perform multiway graph matching on two datasets, CMU [16] and Willow Object Class [20]. CMU is composed of House and Hotel objects viewed under constant illumination and smooth motion. Initial pairwise correspondences as well as ground truth (GT) absolute mappings are provided within the dataset. Object images in Willow dataset include pose, lighting, instance and environment variation as shown in Fig. 3, rendering naive template matching infeasible. For our evaluations, we follow the same design as Wang *et al*. [88]. We first extract local features from a set of 227×227 patches centered around the annotated landmarks, using the prosperous Alexnet [50] pretrained on ImageNet [28]. Our descriptors correspond to the feature map responses of Conv4 and Conv5 layers anchored on the hand annotated keypoints. These features are then matched by the Hungarian algorithm [60] to obtain initial pairwise permutation matrices $\mathbf{P}_0$.

We initialize our algorithm by the closed form MatchEIG [59] and evaluate it against the state of the art methods of Spectral [64], MatchALS [98], MatchLift [42], MatchEIG [59], and Wang *et al*. [88]. The size of the universe is set to the number of features per image. We assume that this number is fixed and partial matches are not present. Handling partialities while using the Birkhoff structure is left as a future work. Note that [88] uses a similar cost function to ours in order to initialize an alternating procedure that in addition exploits the geometry of image coordinates. Authors also use this term as an extra bit of information during their initialization. The standard evaluation metric, *recall*, is defined over the pairwise permutations as:

$$R(\{\hat{\mathbf{P}}_i\}|\mathbf{P}^{\text{gnd}}) = \frac{1}{n|\mathcal{E}|}\sum_{(i,j)\in\mathcal{E}} \mathbf{P}_{ij}^{\text{gnd}} \odot (\hat{\mathbf{P}}_i\hat{\mathbf{P}}_j^\top) \quad (18)$$

where $\mathbf{P}_{ij}^{\text{gnd}}$ are the GT relative transformations and $\hat{\mathbf{P}}_i$ is an estimated permutation. $R = 0$ in the case of no correctly found correspondences and $R = 1$ for a perfect solution. Tab. 1 shows the results of different algorithms as well as ours. Note that our Birkhoff-LRBFGS method that operates solely on pairwise permutations outperforms all methods, even the ones which make use of geometry during initialization. Moreover, when our method is used to initialize Wang *et al*. [88] and perform geometric optimization, we attain the top results. These findings validate that walking on the Birkhoff Polytope, even approximately, and using Riemannian line-search algorithms constitute a promising direction for optimizing the problem at hand.

Uncertainty Estimation in Real Data We now run our confidence estimator on the same Willow Object Class [20].

Table 1. Our results on the *WILLOW Object Class* graph matching dataset. Wang$^-$ refers to running Wang [88] without the geometric consistency term. The vanilla version of our method, *Ours*, already lacks this term. *Ours-Geom* then refers to initializing Wang's verification method with our algorithm. For all the methods, we use the original implementation of the authors.

Dataset	Initial	Spectral [64]	MatchLift [42]	MatchALS [98]	MatchEig [59]	Wang$^-$ [88]	Ours	Wang [88]	Ours-Geom
Car	0.48	0.55	0.65	0.69	0.66	0.72	0.71	1.00	1.00
Duck	0.43	0.59	0.56	0.59	0.56	0.63	0.67	0.93[2]	0.96
Face	0.86	0.92	0.94	0.93	0.93	0.95	0.95	1.00	1.00
Motorbike	0.30	0.25	0.27	0.34	0.28	0.40	0.37	1.00	1.00
Winebottle	0.52	0.64	0.72	0.70	0.71	0.73	0.73	1.00	1.00
CMU-House	0.68	0.90	0.94	0.92	0.94	0.98	0.98	1.00	1.00
CMU-Hotel	0.64	0.81	0.87	0.86	0.92	0.94	0.96	1.00	1.00
Average	0.52	0.59	0.65	0.66	0.71	0.76	**0.77**	0.99	**0.99**

To do that, we first find the optimal point where synchronization is at its best. Then, we set $h \leftarrow 0.0001$, $\beta \leftarrow [0.075, 0.1]$ and automatically start sampling the posterior around this mode for 1000 iterations. Note that β is a critical parameter which can also be dynamically controlled [9]. Larger values of β cannot provide enough variation for a good diversity of solutions. Smaller values cause greater random perturbations leading to samples far from the optimum. This can cause divergence or samples not explaining the local mode. Nevertheless, all our tests worked well with values in the given range.

The generated samples are useful in many applications, e.g. fitting distributions or providing additional solution hints. We address the case of multiple hypotheses generation for the permutation synchronization problem and show that generating an additional per-edge candidate with high certainty helps to improve the recall. Tab. 2 shows the top-K scores we achieve by simply incorporating K likely samples. Note that, when 2 matches are drawn at random and contribute as potentially correct matches, the recall is increased only by 2%, whereas including our samples instead boosts the multi-way matching by 6%.

Table 2. Using **top-K** errors to rank by uncertainty. Based on the confidence information we could retain multiple hypotheses. This is not possible by the other approaches such as Wang *et al*. [59, 88]. *Rand-K* refers to using $K-1$ additional random hypotheses to complement the found solution. *Ours-K* ranks assignments by our probabilistic certainty and retains top-K candidates per point.

Dataset	Wang	Ours	Rand-2	Ours-2	Rand-3	Ours-3
Car	0.72	0.71	0.73	0.76	0.76	0.81
Duck	0.63	0.67	0.67	0.69	0.67	0.72
Face	0.95	0.95	0.96	0.97	0.96	0.98
Motorbike	0.40	0.37	0.45	0.49	0.52	0.60
Winebottle	0.73	0.74	0.77	0.82	0.79	0.85
Avg.	0.69	0.69	0.71	**0.75**	0.74	**0.79**

We further present illustrative results for our confidence prediction in Fig. 4. There, unsatisfactory solutions arising in certain cases are improved by analyzing the uncertainty map. The column (e) of the figure depicts the top-2 assignments retained in the confidence map and (e) plots the assignments that overlap with the true solution. Note that, we might not have access to such an oracle in real applications and only show this to illustrate potential use cases of the estimated confidence map.

[2] [88] reports a value of 0.88, but for their method, we attained 0.93 and therefore report this value.

6.2. Evaluations on Synthetic Data

We synthetically generate 28 different problems with varying sizes: $M \in [10, 100]$ nodes and $n \in [16, 100]$ points in each node. For the scenario of image matching, this would correspond to M cameras and N features in each image. We then introduce $15\% - 35\%$ random swaps to the GT absolute permutations and compute the *observed* relative ones. Details of this dataset are given in suppl. material. Among all 28 sets of synthetic data, we attain an overall recall of 91% whereas MatchEIG [59] remains about 83%.

Runtime Analysis Next, we assess the computational cost of our algorithm against the state of the art methods, on the dataset explained above. All of our experiments are run on a MacBook computer with an Intel i7 2.8GhZ CPU. Our implementation uses a modified Ceres solver [2]. All the other algorithms use highly vectorized MATLAB 2017b code making our comparisons reasonably fair. Fig. 5 tabulates runtimes for different methods excluding initialization. *MatchLift* easily took more than 20min. for moderate problems and hence we choose to exclude it from this evaluation. It is noticeable that thanks to the ability of using more advanced solvers such as LBFGS, our method converges much faster than Wang *et al*. and runs on par with the fastest yet least accurate spectral synchronization [64]. The worst case theoretical computational complexity of our algorithm is $O_{\text{B-LRBFGS}} := O(K|\mathcal{E}|K_S(n^2 + (2n)^3)$ where K is the number of LBFGS iterations and K_S the number of Sinkhorn iterations. While K_S can be a bottleneck, in practice our matrices are already restricted to the Birkhoff manifold and Sinkhorn early-terminates, letting K_S remain small. The complexity is: (1) linearly-dependent upon the

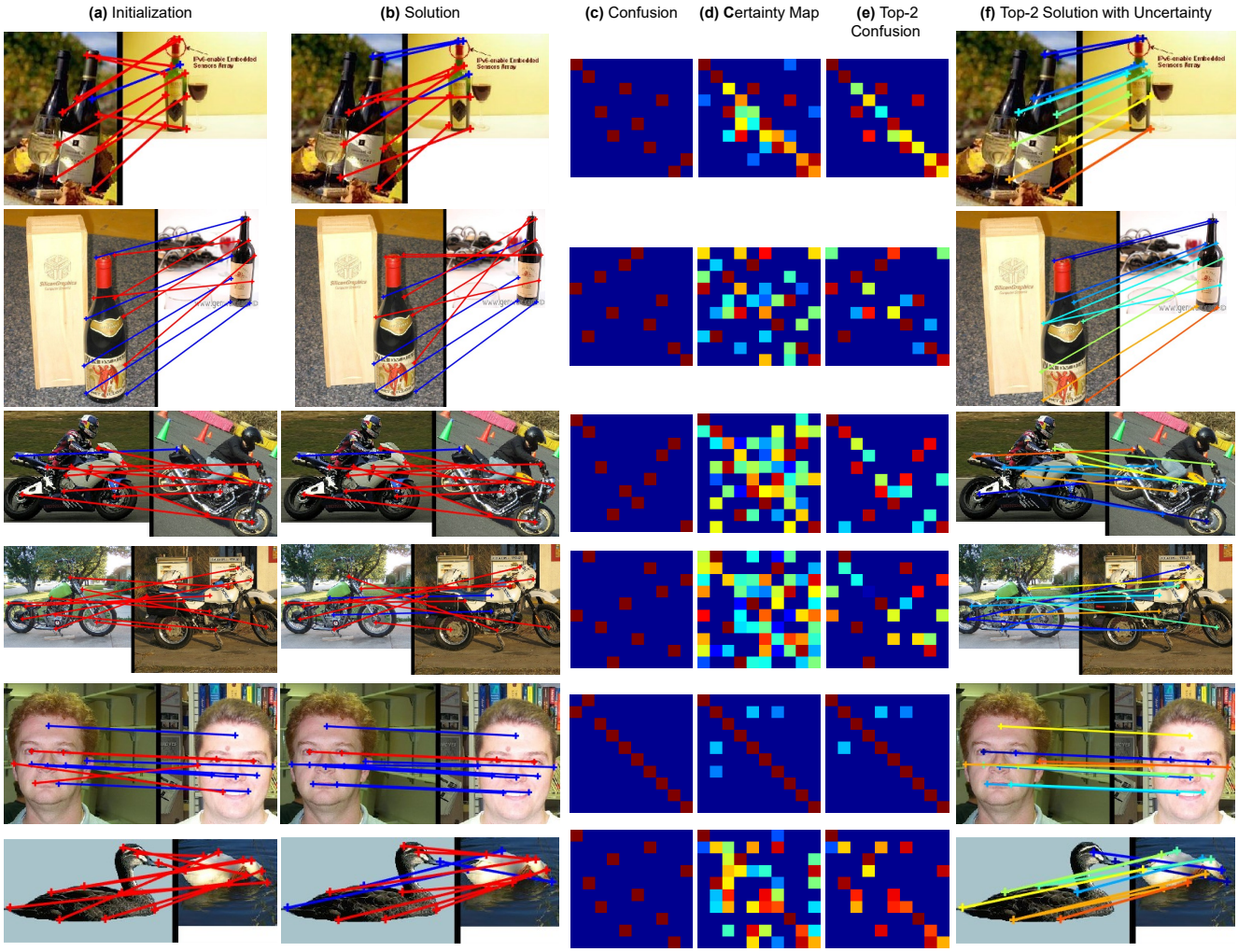


Figure 4. Results from our confidence estimation. Given potentially erroneous solutions (**b**) to the problems initialized as in (**a**), our latent samples discover the uncertain assignments as shown in the middle three columns (**c**-**e**). When multiple top-2 solutions are accepted as potential positives, our method can suggest high quality hypotheses (**f**). The edges in the last column (**f**) is colored by their confidence value. Note that even though, for the sake of space we show pairs of images, the datasets contain multiple sets of images.

number of edges, which in the worst case relates quadratically to the number of images $|\mathcal{E}| = N(N-1)$, (2) cubically dependent on n. This is due to the fact that projection onto the tangent space solves a system of $2n \times 2n$ equations.

7. Conclusion

In this work we have proposed two new frameworks for relaxed permutation synchronization on the manifold of doubly stochastic matrices. Our novel model and formulation paved the way to using sophisticated optimizers such as Riemannian limited-memory BFGS. We further integrated a manifold-MCMC scheme enabling posterior sampling and thereby confidence estimation. We have shown that our confidence maps are informative about cycle inconsistencies and can lead to new solution hypotheses. We used these hypotheses in a top-K evaluation and illustrated its benefits.

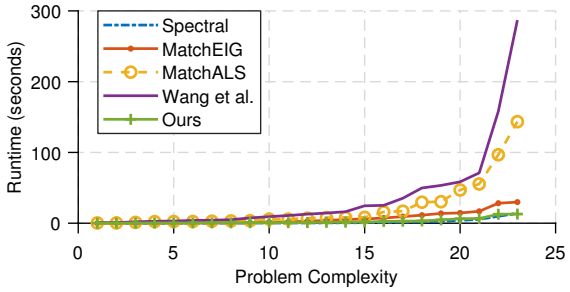


Figure 5. Running times of different methods with increasing problem size: $N \in [10, 100]$ and $n \in [16, 100]$.

In the future, we plan to (i) address partial permutations, the inner region of the Birkhoff Polytope (ii) investigate more sophisticated MCMC schemes such as [32, 39, 74, 54, 76] (iii) seek better use cases for our confidence estimates such as outlier removal.

Acknowledgements Supported by the grant ANR-16-CE23-0014 (FBIMATRIX). Authors thank Haowen Deng for the initial 3D correspondences and, Benjamin Busam and Jesus Briales for fruitful discussions.

References

[1] P-A Absil, Robert Mahony, and Rodolphe Sepulchre. *Optimization algorithms on matrix manifolds*. Princeton University Press, 2009.

[2] Sameer Agarwal, Keir Mierle, and Others. Ceres solver. `http://ceres-solver.org`.

[3] Larry Armijo. Minimization of functions having lipschitz continuous first partial derivatives. *Pacific Journal of mathematics*, 16(1):1–3, 1966.

[4] Federica Arrigoni, Eleonora Maset, and Andrea Fusiello. Synchronization in the symmetric inverse semigroup. In *International Conference on Image Analysis and Processing*, pages 70–81. Springer, 2017.

[5] Federica Arrigoni, Beatrice Rossi, and Andrea Fusiello. Spectral synchronization of multiple views in se (3). *SIAM Journal on Imaging Sciences*, 9(4):1963–1990, 2016.

[6] Florian Bernard, Johan Thunberg, Peter Gemmar, Frank Hertel, Andreas Husch, and Jorge Goncalves. A solution for multi-alignment by transformation synchronisation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2161–2169, 2015.

[7] Florian Bernard, Johan Thunberg, Jorge Goncalves, and Christian Theobalt. Synchronisation of partial multi-matchings via non-negative factorisations. *Pattern Recognition*, 92:146 – 155, 2019.

[8] Tolga Birdal, Emrah Bala, Tolga Eren, and Slobodan Ilic. Online inspection of 3d parts via a locally overlapping camera network. In *2016 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1–10. IEEE, 2016.

[9] Tolga Birdal, Umut Şimşekli, M. Onur Eken, and Slobodan Ilic. Bayesian Pose Graph Optimization via Bingham Distributions and Tempered Geodesic MCMC. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.

[10] Tolga Birdal and Slobodan Ilic. Cad priors for accurate and flexible instance reconstruction. In *Proceedings of the IEEE International Conference on Computer Vision*, 2017.

[11] Garrett Birkhoff. Tres observaciones sobre el algebra lineal. *Univ. Nac. Tucumán Rev. Ser. A*, 5:147–151, 1946.

[12] N. Boumal, B. Mishra, P.-A. Absil, and R. Sepulchre. Manopt, a Matlab toolbox for optimization on manifolds. *Journal of Machine Learning Research*, 15, 2014.

[13] Alexander M Bronstein, Michael M Bronstein, and Ron Kimmel. *Numerical geometry of non-rigid shapes*. Springer Science & Business Media, 2008.

[14] Benjamin Busam, Marco Esposito, Simon Che'Rose, Nassir Navab, and Benjamin Frisch. A stereo vision approach for cooperative robotic movement therapy. In *IEEE International Conference on Computer Vision Workshop (ICCVW)*, December 2015.

[15] Simon Byrne and Mark Girolami. Geodesic monte carlo on embedded manifolds. *Scandinavian Journal of Statistics*, 40(4):825–845, 2013.

[16] Tibério S Caetano, Julian J McAuley, Li Cheng, Quoc V Le, and Alex J Smola. Learning graph matching. *IEEE transactions on pattern analysis and machine intelligence*, 31(6):1048–1058, 2009.

[17] Kunal N Chaudhury, Yuehaw Khoo, and Amit Singer. Global registration of multiple point clouds using semidefinite programming. *SIAM Journal on Optimization*, 25(1):468–501, 2015.

[18] Changyou Chen, Nan Ding, and Lawrence Carin. On the convergence of stochastic gradient MCMC algorithms with high-order integrators. In *Advances in Neural Information Processing Systems*, pages 2278–2286, 2015.

[19] Yuxin Chen, Leonidas Guibas, and Qixing Huang. Near-optimal joint object matching via convex relaxation. In *Proceedings of the 31st International Conference on International Conference on Machine Learning - Volume 32*, ICML'14, pages II–100–II–108. JMLR.org, 2014.

[20] Minsu Cho, Karteek Alahari, and Jean Ponce. Learning graphs to match. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 25–32, 2013.

[21] Stéphan Clémençon and Jérémie Jakubowicz. Kantorovich distances between rankings with applications to rank aggregation. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 2010.

[22] Luca Cosmo, Andrea Albarelli, Filippo Bergamasco, Andrea Torsello, Emanuele Rodolà, and Daniel Cremers. A game-theoretical approach for joint matching of multiple feature throughout unordered images. In *Pattern Recognition (ICPR), 2016 23rd International Conference on*, pages 3715–3720. IEEE, 2016.

[23] Luca Cosmo, Emanuele Rodolà, Andrea Albarelli, Facundo Mémoli, and Daniel Cremers. Consistent par-

tial matching of shape collections via sparse modeling. In *Computer Graphics Forum*, volume 36, pages 209–221. Wiley Online Library, 2017.

[24] Angela Dai, Matthias Nießner, Michael Zollhöfer, Shahram Izadi, and Christian Theobalt. Bundlefusion: Real-time globally consistent 3d reconstruction using on-the-fly surface reintegration. *ACM Transactions on Graphics (TOG)*, 36(4):76a, 2017.

[25] Joseph DeGol, Timothy Bretl, and Derek Hoiem. Improved structure from motion using fiducial marker matching. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 273–288, 2018.

[26] Haowen Deng, Tolga Birdal, and Slobodan Ilic. Ppf-foldnet: Unsupervised learning of rotation invariant 3d local descriptors. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 602–618, 2018.

[27] Haowen Deng, Tolga Birdal, and Slobodan Ilic. Ppfnet: Global context aware local features for robust 3d point matching. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[28] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *IEEE Conference on Computer Vision and Pattern Recognition*. Ieee, 2009.

[29] Persi Diaconis, Susan Holmes, Mehrdad Shahshahani, et al. Sampling from a manifold. In *Advances in Modern Statistical Theory and Applications: A Festschrift in honor of Morris L. Eaton*. Institute of Mathematical Statistics, 2013.

[30] A. Douik and B. Hassibi. Manifold Optimization Over the Set of Doubly Stochastic Matrices: A Second-Order Geometry. *ArXiv e-prints*, Feb. 2018.

[31] Fanny Dufossé, Kamer Kaya, Ioannis Panagiotas, and Bora Uçar. *Further notes on Birkhoff-von Neumann decomposition of doubly stochastic matrices*. PhD thesis, Inria-Research Centre Grenoble–Rhône-Alpes, 2017.

[32] Alain Durmus, Umut Simsekli, Eric Moulines, Roland Badeau, and Gaël Richard. Stochastic gradient Richardson-Romberg Markov Chain Monte Carlo. In *Advances in Neural Information Processing Systems*, pages 2047–2055, 2016.

[33] Rizal Fathony, Sima Behpour, Xinhua Zhang, and Brian Ziebart. Efficient and consistent adversarial bipartite matching. In *International Conference on Machine Learning*, pages 1456–1465, 2018.

[34] Roger Fletcher. *Practical methods of optimization*. John Wiley & Sons, 2013.

[35] P. Gargallo. Using opensfm. Online. Accessed May-2018.

[36] Mark Girolami and Ben Calderhead. Riemann manifold Langevin and Hamiltonian Monte Carlo methods. *Journal of the Royal Statistical Society: Series B 118 (Statistical Methodology)*, 73(2):123–214, 2011.

[37] Michel X Goemans. Smallest compact formulation for the permutahedron. *Mathematical Programming*, 2015.

[38] Venu Madhav 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. IEEE, 2004.

[39] Andrew Holbrook. Note on the geodesic monte carlo. *arXiv preprint arXiv:1805.05289*, 2018.

[40] Seyedehsomayeh Hosseini, Wen Huang, and Rohollah Yousefpour. Line search algorithms for locally lipschitz functions on riemannian manifolds. *SIAM Journal on Optimization*, 28(1):596–619, 2018.

[41] Nan Hu, Qixing Huang, Boris Thibert, UG Alpes, and Leonidas Guibas. Distributable consistent multi-object matching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[42] Qi-Xing Huang and Leonidas Guibas. Consistent shape maps via semidefinite programming. In *Proceedings of the Eleventh Eurographics/ACMSIGGRAPH Symposium on Geometry Processing*. Eurographics Association, 2013.

[43] Wen Huang, P.-A. Absil, K. A. Gallivan, and Paul Hand. Roptlib: an object-oriented c++ library for optimization on riemannian manifolds. Technical Report FSU16-14.v2, Florida State University, 2016.

[44] Wen Huang, Kyle A Gallivan, and P-A Absil. A broyden class of quasi-newton methods for riemannian optimization. *SIAM Journal on Optimization*, 25(3):1660–1685, 2015.

[45] Daniel F Huber and Martial Hebert. Fully automatic registration of multiple 3d data sets. *Image and Vision Computing*, 21(7):637–650, 2003.

[46] GLENN Hurlbert. A short proof of the birkhoff-von neumann theorem. *preprint (unpublished)*, 2008.

[47] Bruno Iannazzo and Margherita Porcelli. The riemannian barzilai–borwein method with nonmonotone line search and the matrix geometric mean computation. *IMA Journal of Numerical Analysis*, 38(1):495–517, 2017.

[48] Ross Kindermann. Markov random fields and their applications. *American mathematical society*, 1980.

[49] Anna Korba, Alexandre Garcia, and Florence d'Alché Buc. A structured prediction approach for label ranking. In *Advances in Neural Information Processing Systems*, pages 8994–9004, 2018.

[50] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[51] Hongdong Li and Richard Hartley. The 3d-3d registration problem revisited. In *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pages 1–8. IEEE, 2007.

[52] Xinchao Li, Martha Larson, and Alan Hanjalic. Pairwise geometric matching for large-scale object retrieval. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5153–5161, 2015.

[53] Scott Linderman, Gonzalo Mena, Hal Cooper, Liam Paninski, and John Cunningham. Reparameterizing the birkhoff polytope for variational permutation inference. In *International Conference on Artificial Intelligence and Statistics*, pages 1618–1627, 2018.

[54] Chang Liu, Jun Zhu, and Yang Song. Stochastic gradient geodesic mcmc methods. In *Advances in Neural Information Processing Systems*, pages 3009–3017, 2016.

[55] David G Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004.

[56] Vince Lyzinski, Donniell E Fishkind, Marcelo Fiori, Joshua T Vogelstein, Carey E Priebe, and Guillermo Sapiro. Graph matching: Relax at your own risk. *IEEE transactions on pattern analysis and machine intelligence*, 38(1):60–73, 2016.

[57] Yi-An Ma, Tianqi Chen, and Emily Fox. A complete recipe for stochastic gradient MCMC. In *Advances in Neural Information Processing Systems*, pages 2917–2925, 2015.

[58] João Maciel and João Paulo Costeira. A global solution to sparse correspondence problems. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(2), 2003.

[59] E. Maset, F. Arrigoni, and A. Fusiello. Practical and efficient multi-view matching. In *2017 IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[60] James Munkres. Algorithms for the assignment and transportation problems. *Journal of the society for industrial and applied mathematics*, 5(1):32–38, 1957.

[61] Raúl Mur-Artal and Juan D. Tardós. ORB-SLAM2: an open-source SLAM system for monocular, stereo and RGB-D cameras. *IEEE Transactions on Robotics*, 33(5):1255–1262, 2017.

[62] Andy Nguyen, Mirela Ben-Chen, Katarzyna Welnicka, Yinyu Ye, and Leonidas Guibas. An optimization approach to improving collections of shape maps. In *Computer Graphics Forum*, volume 30, pages 1481–1491. Wiley Online Library, 2011.

[63] Deepti Pachauri, Risi Kondor, Gautam Sargur, and Vikas Singh. Permutation diffusion maps (pdm) with application to the image association problem in computer vision. In *Advances in Neural Information Processing Systems*, pages 541–549, 2014.

[64] Deepti Pachauri, Risi Kondor, and Vikas Singh. Solving the multi-way matching problem by permutation synchronization. In *Advances in neural information processing systems*, pages 1860–1868, 2013.

[65] Han-Mu Park and Kuk-Jin Yoon. Consistent multiple graph matching with multi-layer random walks synchronization. *Pattern Recognition Letters*, 2018.

[66] Sam Patterson and Yee Whye Teh. Stochastic gradient Riemannian Langevin dynamics on the probability simplex. In *Advances in Neural Information Processing Systems*, pages 3102–3110, 2013.

[67] Sergey Plis, Stephen McCracken, Terran Lane, and Vince Calhoun. Directional statistics on permutations. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, pages 600–608, 2011.

[68] Chunhong Qi, Kyle A Gallivan, and P-A Absil. Riemannian bfgs algorithm with applications. In *Recent advances in optimization and its applications in engineering*, pages 183–192. Springer, 2010.

[69] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 652–660, 2017.

[70] Wolfgang Ring and Benedikt Wirth. Optimization methods on riemannian manifolds and their application to shape space. *SIAM Journal on Optimization*, 22(2):596–627, 2012.

[71] Torsten Sattler, Tobias Weyand, Bastian Leibe, and Leif Kobbelt. Image retrieval for image-based localization revisited, 2012.

[72] Michele Schiavinato and Andrea Torsello. Synchronization over the birkhoff polytope for multi-graph matching. In *International Workshop on Graph-Based Representations in Pattern Recognition*, pages 266–275. Springer, 2017.

[73] Johannes L Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Proceedings of*

the IEEE Conference on Computer Vision and Pattern Recognition, pages 4104–4113, 2016.

[74] Umut Şimşekli. Fractional Langevin Monte Carlo: Exploring Lévy driven stochastic differential equations for Markov Chain Monte Carlo. In *International Conference on Machine Learning*, pages 3200–3209. JMLR. org, 2017.

[75] Umut Simsekli, Roland Badeau, Taylan Cemgil, and Gaël Richard. Stochastic quasi-Newton Langevin Monte Carlo. In *International Conference on Machine Learning (ICML)*, 2016.

[76] Umut Şimşekli, Çağatay Yıldız, Thanh Huy Nguyen, Gaël Richard, and A Taylan Cemgil. Asynchronous stochastic quasi-Newton MCMC for non-convex optimization. In *International Conference on Machine Learning*, 2018.

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

[78] Amit Singer and Yoel Shkolnisky. Three-dimensional structure determination from common lines in cryo-em by eigenvectors and semidefinite programming. *SIAM journal on imaging sciences*, 4(2):543–572, 2011.

[79] Richard Sinkhorn and Paul Knopp. Concerning nonnegative matrices and doubly stochastic matrices. *Pacific Journal of Mathematics*, 21(2):343–348, 1967.

[80] Steven T Smith. Optimization techniques on Riemannian manifolds. *Fields institute communications*, 3(3):113–135, 1994.

[81] Yifan Sun, Zhenxiao Liang, Xiangru Huang, and Qixing Huang. Joint map and symmetry synchronization. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 251–264, 2018.

[82] Da Tang and Tony Jebara. Initialization and coordinate optimization for multi-way matching. In *Artificial Intelligence and Statistics*, pages 1385–1393, 2017.

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

[84] Bill Triggs, Philip F McLauchlan, Richard I Hartley, and Andrew W Fitzgibbon. Bundle adjustment—a modern synthesis. In *International workshop on vision algorithms*, pages 298–372. Springer, 1999.

[85] Nilesh Tripuraneni, Nicolas Flammarion, Francis Bach, and Michael I Jordan. Averaging stochastic gradient descent on Riemannian manifolds. In *Conference on Learning Theory (COLT)*, 2018.

[86] Roberto Tron and Rene Vidal. Distributed 3-d localization of camera sensor networks from 2-d image measurements. *IEEE Transactions on Automatic Control*, 59(12):3325–3340, 2014.

[87] Lanhui Wang and Amit Singer. Exact and stable recovery of rotations for robust synchronization. *Information and Inference: A Journal of the IMA*, 2(2):145–193, 2013.

[88] Qianqian Wang, Xiaowei Zhou, and Kostas Daniilidis. Multi-image semantic matching by mining consistent features. In *CVPR*, 2018.

[89] Philip Wolfe. Convergence conditions for ascent methods. *SIAM review*, 11(2):226–235, 1969.

[90] Philip Wolfe. Convergence conditions for ascent methods. ii: Some corrections. *SIAM review*, 13(2):185–188, 1971.

[91] Tatiana Xifara, Chris Sherlock, Samuel Livingstone, Simon Byrne, and Mark Girolami. Langevin diffusions and the Metropolis-adjusted Langevin algorithm. *Statistics & Probability Letters*, 91:14–19, 2014.

[92] Junchi Yan, Minsu Cho, Hongyuan Zha, Xiaokang Yang, and Stephen M Chu. Multi-graph matching via affinity optimization with graduated consistency regularization. *IEEE transactions on pattern analysis and machine intelligence*, 38(6):1228–1242, 2016.

[93] Junchi Yan, Xu-Cheng Yin, Weiyao Lin, Cheng Deng, Hongyuan Zha, and Xiaokang Yang. A short survey of recent advances in graph matching. In *Proceedings of the 2016 ACM on International Conference on Multimedia Retrieval*, pages 167–174. ACM, 2016.

[94] Jin-Gang Yu, Gui-Song Xia, Ashok Samal, and Jinwen Tian. Globally consistent correspondence of multiple feature sets using proximal gauss–seidel relaxation. *Pattern Recognition*, 51:255–267, 2016.

[95] Xinru Yuan, Wen Huang, P-A Absil, and Kyle A Gallivan. A riemannian limited-memory BFGS algorithm for computing the matrix geometric mean. *Procedia Computer Science*, 80:2147–2157, 2016.

[96] Andrei Zanfir and Cristian Sminchisescu. Deep learning of graph matching. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[97] Mikhail Zaslavskiy, Francis Bach, and Jean-Philippe Vert. A path following algorithm for the graph matching problem. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(12):2227–2242, 2009.

[98] Xiaowei Zhou, Menglong Zhu, and Kostas Daniilidis. Multi-image matching via fast alternating minimization. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4032–4040, 2015.

Probabilistic Permutation Synchronization using the Riemannian Structure of the Birkhoff Polytope - Supplementary Material

This part supplements our main paper by providing further algorithmic details on RL-BFGS, proofs of the propositions presented in the paper, derivations of the *retraction Euler integrator* and additional experiments.

A. Riemannian Limited Memory BFGS

We now explain the LR-BFGS optimizer used in our work. To begin with, we recall the foundational BFGS [34], that is a quasi Newton method operating in the Euclidean space. We then review the simple Riemannian descent algorithms that employ line-search. Finally, we explain the R-BFGS used to solve our synchronization problem. R-BFGS can be modified slightly to arrive at the desired limited memory Riemannian BFGS solver.

A.1. Euclidean BFGS

The idea is to approximate the true hessian $\mathbf{H}$ with $\boldsymbol{\mathcal{B}}$, using updates specified by the gradient evaluations[3]. We will then transition from this Euclidean space line search method to Riemannian space optimizers. For clarity, in Alg. 1 we summarize the Euclidean BFGS algorithm, that computes $\boldsymbol{\mathcal{B}}$ by using the most recent values of the past iterates. Note that many strategies exist to initialize $\boldsymbol{\mathcal{B}}_0$, while a common choice is the scaled identity matrix $\boldsymbol{\mathcal{B}}_0 = \gamma\mathbf{I}$. Eq. 19 corresponds to the particular BFGS-update rule. In the limited-memory successor of BFGS, the L-BFGS, the Hessian matrix $\mathbf{H}$ is instead approximated up to a pre-specified rank in order to achieve linear time and space-complexity in the dimension of the problem.

Algorithm 1: Euclidean BFGS

input: A real-valued, differentiable potential energy U, initial iterate $\mathbf{X}_0$ and initial Hessian approximation $\boldsymbol{\mathcal{B}}_0$.
$k \leftarrow 0$
while $\mathbf{x}_k$ *does not sufficiently minimize* f **do**
  Compute the direction $\boldsymbol{\eta}_k$ by solving $\boldsymbol{\mathcal{B}}_k\boldsymbol{\eta}_k = -\nabla U(\mathbf{x}_k)$ for $\boldsymbol{\eta}_k$.
  Define the new iterate $\mathbf{x}_{k+1} \leftarrow \mathbf{x}_k + \boldsymbol{\eta}_k$.
  Set $\mathbf{s}_k \leftarrow \mathbf{x}_{k+1} - \mathbf{x}_k$ and $\mathbf{y}_k \leftarrow \nabla U(\mathbf{x}_{k+1}) - \nabla U(\mathbf{x}_k)$.
  Compute the new Hessian approximation:

$$\boldsymbol{\mathcal{B}}_{k+1} = \boldsymbol{\mathcal{B}}_k + \frac{\mathbf{y}_k\mathbf{y}_k^\top}{\mathbf{y}_k^\top\mathbf{s}_k} - \frac{\boldsymbol{\mathcal{B}}_k\mathbf{s}_k\mathbf{s}_k^\top\boldsymbol{\mathcal{B}}_k}{\mathbf{s}_k^\top\boldsymbol{\mathcal{B}}_k\mathbf{s}_k}. \tag{19}$$

  $k \leftarrow k + 1$.

A.2. Riemannian Descent and Line Search

The standard descent minimizers can be extended to operate on Riemannian manifolds $\mathcal{M}$ using the geometry of the parameters. A typical Riemannian update can be characterized as:

$$\mathbf{x}_{k+1} = R_{\mathbf{x}_k}(\tau_k\boldsymbol{\eta}_k) \tag{20}$$

where R is the retraction operator, i.e. the smooth map from the tangent bundle $\mathcal{TM}$ to $\mathcal{M}$ and $R_{\mathbf{x}_k}$ is the restriction of R to $\mathcal{T}_{\mathbf{x}_k}$, the tangent space of the current iterate $\mathbf{x}_k$, $R_{\mathbf{x}_k} : \mathcal{T}_{\mathbf{x}_k} \to \mathcal{M}$. The descent direction is defined to be on the tangent space $\boldsymbol{\eta}_k \in \mathcal{T}_{\mathbf{x}_k}\mathcal{M}$. When manifolds are rather simple shapes, the size of the step τ_k can be a fixed value. However, for most matrix manifolds, some form of a line search is preferred to compute τ_k. The retraction operator is used to take steps on the manifold and is usually derived analytically. When this analytic map is length-preserving, it is called *true* exponential map. However, such exactness is not a requirement for the optimizer and as it happens in the case of doubly stochastic matrices, $R_{\mathbf{x}_k}$ only needs to be an approximate *retraction*, e.g. first or second order. In fact, for a map to be valid retraction, it is sufficient to satisfy the following conditions:

[3]Note that certain implementations can instead opt to approximate the inverse Hessian for computational reasons.

1. R is continuously differentiable.
2. $R_{\mathbf{x}}(\mathbf{0}_{\mathbf{x}}) = \mathbf{x}$, where $\mathbf{0}_{\mathbf{x}}$ denotes the zero element of $\mathcal{T}_{\mathbf{x}}\mathcal{M}$. This is called the *centering* property.
3. The curve $\gamma_{\boldsymbol{\eta}_{\mathbf{x}}}(\tau) = R_{\mathbf{x}}(\tau\boldsymbol{\eta}_{\mathbf{x}})$ satisfies:

$$\frac{d\gamma_{\boldsymbol{\eta}_{\mathbf{x}}}(\tau)}{d\tau}\Big|_{\tau=0} = \boldsymbol{\eta}_{\mathbf{x}},\ \forall \boldsymbol{\eta}_{\mathbf{x}} \in \mathcal{T}_{\mathbf{x}}\mathcal{M}. \tag{21}$$

This is called the *local rigidity* property.

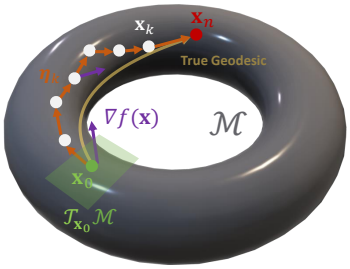


Figure 6. Visualization of the entities used on a sample toroidal manifold.

Considering all these, we provide, in Alg. 2, a general Riemannian descent algorithm, which can be customized by the choice of the direction, retraction and the means to compute the step size. Such a concept of minimizing by walking on the manifold is visualized in Fig. 6.

Algorithm 2: General Riemannian Line Search Minimizer

input: A Riemannian manifold $\mathcal{M}$, a retraction operator R and initial iterate $\mathbf{x}_k \in \mathcal{M}$ where $k = 0$.
while $\mathbf{x}_k$ *does not sufficiently minimize* f **do**
  Pick a gradient related descent direction $\boldsymbol{\eta}_k \in \mathcal{T}_{\mathbf{x}_k}\mathcal{M}$.
  Choose a retraction $R_{\mathbf{x}_k} : \mathcal{T}_{\mathbf{x}_k}\mathcal{M} \to \mathcal{M}$.
  Choose a step length $\tau_k \in \mathbb{R}$.
  Set $\mathbf{x}_{k+1} \leftarrow R_{\mathbf{x}_k}(\tau_k\boldsymbol{\eta}_k)$.
  $k \leftarrow k + 1$.

It is possible to develop new minimizers by making particular choices for the Riemannian operators in Alg 2. We now review the Armijo variant of the Riemannian gradient descent [40], that is a common and probably the simplest choice for an accelerated optimizer on the manifold. Though, many other line-search conditions such as Barzilai-Borwein [47] or strong Wolfe [70] can be used. The pseudocode for this backtracking version is given in Alg. 3. Note that the Riemannian gradient $\text{grad}f(\mathbf{x})$ is simply obtained by projecting the Euclidean gradient $\nabla f(\mathbf{x})$ onto the manifold $\mathcal{M}$ and the next iterate is obtained through a line search so as to satisfy the Armijo condition [3], tweaked to use the retraction R for taking steps. For some predefined Armijo step size, this algorithm is guaranteed to converge for all retractions [1].

Algorithm 3: General Riemannian Steepest Descent with Armijo Line Search

input: A Riemannian manifold $\mathcal{M}$, a retraction operator R, the projection operator onto the tangent space $\Pi_{\mathbf{x}_k} : \mathbb{R}^n \to \mathcal{T}_{\mathbf{x}_k}\mathcal{M}$, a real-valued, differentiable potential energy f, initial iterate $\mathbf{x}_0 \in \mathcal{M}$ and the Armijo line search scalars including c.
while $\mathbf{x}_k$ *does not sufficiently minimize* f **do**
  `// Euclidean gradient to Riemannian direction`
  $\boldsymbol{\eta}_k \leftarrow -\text{grad}f(\mathbf{x}_k) \triangleq \Pi_{\mathbf{x}_k}(-\nabla f(\mathbf{x}_k))$.
  Select $\mathbf{x}_{k+1}$ such that:

$$f(\mathbf{x}_k) - f(\mathbf{x}_{k+1}) \geq c\big(f(\mathbf{x}_k) - f(R_{\mathbf{x}}(\tau_k\boldsymbol{\eta}_k))\big), \tag{22}$$

  where τ_k is the Armijo step size.
  $k \leftarrow k + 1$.

A.3. LR-BFGS for Minimization on $\mathcal{DP}_n$

Based upon the ideas developed up to this point, we present the Riemannian variant of the L-BFGS. The algorithm is mainly based on Huang *et al*. and we refer the reader to their seminal work for more details [44]. It is also worth noting [68]. Similar to the Euclidean case, we begin by summarizing a BFGS algorithm with the difference that it is suited to solving the synchronization task. This time, $\boldsymbol{\mathcal{B}}$ will approximate the action of the Hessian on the tangent space $\mathcal{T}_{\mathbf{x}_k}\mathcal{M}$. Generalizing any (quasi-)Newton method to the Riemannian setting thus requires computing the Riemannian Hessian operator, or its approximation. This necessitates taking a some sort of a directional derivative of a vector field. As the vector fields belonging

to different tangent spaces cannot be compared immediately, one needs the notion of connection Γ generalizing the directional derivative of a vector field. This connection is closely related to the concept of vector transport $T : \mathcal{TM} \otimes \mathcal{TM} \to \mathcal{TM}$ which allows moving from one tangent space to the other $T_{\boldsymbol{\eta}}(\boldsymbol{\zeta}) : (\boldsymbol{\eta}, \boldsymbol{\zeta}) \in \mathcal{T}_{\mathbf{x}}\mathcal{M} \to \mathcal{T}_{R_{\mathbf{x}}(\boldsymbol{\eta})}\mathcal{M}$. For the first order treatment of the Birkhoff Polytope, this vector transport takes a simple form:

$$T_{\boldsymbol{\eta}}(\zeta) \in \mathcal{T}_{R_{\mathbf{x}}(\boldsymbol{\eta})}\mathcal{DP}_n \triangleq \Pi_{R_{\mathbf{x}}(\boldsymbol{\eta})}(\boldsymbol{\zeta}). \tag{23}$$

where $\Pi_{\mathbf{X}}(\cdot)$ is the projection onto the tangent space of $\mathbf{X}$ as defined in the main paper. We give the pseudocode of the R-BFGS algorithm in Alg. 4 below. To ensure Riemannian L-BFGS always produces a descent direction, it is necessary to adapt a line-search algorithm which satisfies strong Wolfe (SW) conditions [89, 90]. Roptlib [43] does implement the SW while ManOpt [12] uses the simpler Armijo conditions, even for the LR-BFGS. Another simple possibility is to take the steps on the Manifold using the retraction, while performing the line search in the Euclidean space. This results in a projected, approximate line search, but can terminate quite quickly, making it possible to exploit existing Euclidean space SW LBFGS solvers such as Ceres [2]. Without delving into rigorous proofs, we provide one such implementation at `github.com/tolgabirdal/MatrixManifoldsInCeres` where several matrix manifolds are considered. We leave the analysis of such approximations for a future study and note that the results in the paper are generated by limited memory form of the R-BFGS procedure summarized under Alg. 4. While many modifications do exist, LR-BFGS, in essence, is an approximation to R-BFGS, where $O(MN^2)$ storage of the full Hessian is avoided by unrolling the RBFGS descent computation and using the last m input and gradient differences where $m << MN^2$. This allows us to handle large data regimes.

Algorithm 4: Riemannian BFGS for Synchronization on $\mathcal{DP}_n$

input: Birkhoff Polytope $\mathcal{DP}$ with Riemannian (Fisher information) metric g, first order retraction R, the parallel transport T, a real-valued, differentiable potential energy function of synchronization U, initial iterate $\mathbf{X}_0$ and initial Hessian approximation $\boldsymbol{\mathcal{B}}_0$.

while $\mathbf{x}_k$ *does not sufficiently minimize* f **do**

Compute the Euclidean gradients using:

$$\nabla_{\mathbf{X}_i} U(\mathbf{X}) = \sum_{(i,j)\in E} -2(\mathbf{P}_{ij} - \mathbf{X}_i\mathbf{X}_j^\top)\mathbf{X}_j \qquad \nabla_{\mathbf{X}_j} U(\mathbf{X}) = \sum_{(i,j)\in E} -2(\mathbf{P}_{ij} - \mathbf{X}_i\mathbf{X}_j^\top)\mathbf{X}_i \tag{24}$$

Compute the Riemannian gradient $\text{grad}\, U(\mathbf{X}_k) \leftarrow\!\!\triangleq \Pi_{\mathbf{X}_k}(-\nabla U(\mathbf{X}_k))$.

Compute the direction $\boldsymbol{\eta}_k \in \mathcal{T}_{\mathbf{X}_k}\mathcal{DP}_n$ by solving $\boldsymbol{\mathcal{B}}_k\boldsymbol{\eta}_k = -\text{grad}\, U(\mathbf{X}_k)$.

Given $\boldsymbol{\eta}_k$, execute a line search satisfying the strong Wolfe conditions [89, 90, 44]. Set step size τ_k.

Set $\mathbf{x}_{k+1} = R_{\mathbf{x}_k}(\tau_k\boldsymbol{\eta}_k)$.

Use vector transport to define:

$$\mathbf{S}_k = T_{\tau_k\boldsymbol{\eta}_k}(\tau_k\boldsymbol{\eta}_k)\,, \qquad \mathbf{Y}_k = \text{grad}\, U(\mathbf{X}_{k+1}) - T_{\tau_k\boldsymbol{\eta}_k}(\text{grad}\, U(\mathbf{x}_k)). \tag{25}$$

Compute $\tilde{\boldsymbol{\mathcal{B}}_k} = T_{\tau_k\boldsymbol{\eta}_k} \circ \boldsymbol{\mathcal{B}}_k \circ T^{+}_{\tau_k\boldsymbol{\eta}_k}$ where T^+ is denotes (pseudo-)inverse of the transport.

Compute the linear operator $\boldsymbol{\mathcal{B}}_{k+1} : \mathcal{T}_{\mathbf{x}_{k+1}}\mathcal{M} \to \mathcal{T}_{\mathbf{x}_{k+1}}\mathcal{M}$:

$$\tilde{\boldsymbol{\mathcal{B}}}_{k+1}\mathbf{Z} = \tilde{\boldsymbol{\mathcal{B}}}_k\mathbf{Z} + \frac{g(\mathbf{Y}_k, \mathbf{Z})}{g(\mathbf{y}_k, \mathbf{S}_k)}\mathbf{Y}_k - \frac{g(\mathbf{S}_k, \tilde{\boldsymbol{\mathcal{B}}}\mathbf{Z})}{g(\mathbf{S}_k, \tilde{\boldsymbol{\mathcal{B}}}\mathbf{S}_k)} \qquad \forall \mathbf{Z} \in \mathcal{T}_{\mathbf{x}_{k+1}}\mathcal{DP}_n. \tag{26}$$

$k \leftarrow k + 1$.

B. Proof of Proposition 1 and Further Analysis

Proof. Consider the ray cast outwards from the origin: $\mathbf{r} = \mathbb{R}_{\geq 0}\mathbb{1}$. $\mathbf{r}$ exits $\mathcal{DP}_n$ at $\frac{1}{n}\mathbb{1}$ but exits the sphere $\mathcal{S}^{n-1}$ at $\frac{1}{\sqrt{n}}\mathbb{1}$. This creates a gap between the two manifolds whose ratio grows to ∞ as n grows. For a perspective of optimization, consider the linear function $\lambda(\mathbf{P}) = \sum_{i,j} P_{ij}$. $\lambda(\mathbf{P})$ is minimized on $\mathcal{DP}_n$ at $\frac{1}{n}\mathbb{1}$ and on the sphere at $\frac{1}{\sqrt{n}}\mathbb{1}$, □

We now look at the restricted orthogonal matrices and seek to find the difference between optimizing a linear functional on them and $\mathcal{DP}_n$. Note that minimizing functionals is ultimately what we are interested in as the problems we consider here are formulated as optimization. Let $\mathcal{A}$ be the affine linear space of $(n-1)\times(n-1)$ matrices whose rows and columns sum to 1, *i.e*. doubly stochastic but with no condition on the signs or *a generalized doubly stochastic matrix*. The Birkhoff Polytope $\mathcal{DP}_n$ is contained in $\mathcal{A}$, whereas the orthogonal group $\mathcal{O}_n$ is not. So, there exists an affine functional λ that is minimized at a point on $\mathcal{DP}_n$, $\lambda = 0$ but not on $\mathcal{O}$. Let $\mathcal{AO}_n = \mathcal{DP}_n \cap \mathcal{O}_n$, a further restricted manifold. This time unlike the case of $\mathcal{DP}_n$, $\mathcal{AO}_n$ would not coincide the permutations $\mathcal{P}$ due to the negative elements. In fact $\mathcal{P} \subset \mathcal{AO}_n$.

Proposition 3. *The ratio between the time a ray leaves $\mathcal{DP}_n$ and the same ray leaves $\mathcal{AO}_n$ can be as large as $n-1$.*

Proof. Consider the line through $\frac{1}{n}\mathbb{1}$ and $\mathbf{I}$: $l(x) = \frac{1+x}{n}\mathbb{1} - x\mathbf{I}$. Such a ray leaves $\mathcal{DP}_n$ at $x = \frac{1}{n-1}$ and for all $x \in [-1, 1]$ is in the convex hull of $\mathcal{AO}_n$. When $x = 1$, it is an orthogonal matrix, *i.e*. on $\mathcal{O}_n$. Hence, the ray can be on $\mathcal{AO}_n$ for $n-1$ times as long as in $\mathcal{DP}_n$. This quantity also grows to infinity as $n \to \infty$. If same analysis is done for the case of the sphere, the ratio is found to be $(n-1)^{3/2}$, grows quicker to infinity and is a larger quantity. □

C. Proof of Proposition 2

Proof. The conclusion of the proposition states that $p(\mathbf{X})$ and $p(\mathbf{P}|\mathbf{X})$ have the following form:

$$p(\mathbf{X}) = \frac{1}{C}\exp\Big(-\beta \sum_{(i,j)\in\mathcal{E}} \|\mathbf{X}_{ij}\|_{\mathrm{F}}^2\Big) \prod_{(i,j)\in\mathcal{E}} Z_{ij} \tag{27}$$

$$p(\mathbf{P}|\mathbf{X}) = \prod_{(i,j)\in\mathcal{E}} p(\mathbf{P}_{ij}|\mathbf{X}_i, \mathbf{X}_j) \tag{28}$$

$$= \prod_{(i,j)\in\mathcal{E}} \exp\Big(2\beta\,\mathrm{tr}(\mathbf{P}_{ij}^\top \mathbf{X}_{ij})\Big)\frac{1}{Z_{ij}} \tag{29}$$

$$= \exp\Big(\beta \sum_{(i,j)\in\mathcal{E}} 2\mathrm{tr}(\mathbf{P}_{ij}^\top \mathbf{X}_{ij})\Big) \prod_{(i,j)\in\mathcal{E}} \frac{1}{Z_{ij}} \tag{30}$$

where

$$Z_{ij} := \prod_{b=1}^{B_{ij}} Z_{\mathbf{X}}(\beta, \theta_{ij,b}). \tag{31}$$

Our goal is to verify that the following holds with the above definitions:

$$p(\mathbf{P}, \mathbf{X}) = \frac{1}{Z}\exp\Big(-\beta \sum_{(i,j)\in\mathcal{E}} \|\mathbf{P}_{ij} - \mathbf{X}_i\mathbf{X}_j^\top\|_{\mathrm{F}}^2\Big) = \frac{1}{Z}\prod_{(i,j)\in\mathcal{E}} \psi(\mathbf{P}_{ij}, \mathbf{X}_i, \mathbf{X}_j), \tag{32}$$

where Z is a positive constant (cf. Section 4 in the main paper). Then, we can easily verify this equality as follows:

$$p(\mathbf{P}, \mathbf{X}) = p(\mathbf{P}|\mathbf{X})p(\mathbf{X}) \tag{33}$$

$$= \frac{1}{C}\exp\Big(-\beta \sum_{(i,j)\in\mathcal{E}} \|\mathbf{X}_{ij}\|_{\mathrm{F}}^2\Big)\exp\Big(\beta \sum_{(i,j)\in\mathcal{E}} 2\mathrm{tr}(\mathbf{P}_{ij}^\top \mathbf{X}_{ij})\Big) \tag{34}$$

$$= \frac{1}{C}\exp\Big(-\beta \sum_{(i,j)\in\mathcal{E}} \big(\|\mathbf{X}_{ij}\|_{\mathrm{F}}^2 - 2\mathrm{tr}(\mathbf{P}_{ij}^\top \mathbf{X}_{ij})\big)\Big) \tag{35}$$

$$= \frac{1}{C}\exp\Big(-\beta \sum_{(i,j)\in\mathcal{E}} \big(\|\mathbf{X}_{ij}\|_{\mathrm{F}}^2 - 2\mathrm{tr}(\mathbf{P}_{ij}^\top \mathbf{X}_{ij}) - n + n\big)\Big) \tag{36}$$

$$= \frac{1}{C}\exp(\beta n|\mathcal{E}|)\exp\Big(-\beta \sum_{(i,j)\in\mathcal{E}} \big(\|\mathbf{X}_{ij}\|_{\mathrm{F}}^2 - 2\mathrm{tr}(\mathbf{P}_{ij}^\top \mathbf{X}_{ij}) + \|\mathbf{P}_{ij}\|_{\mathrm{F}}^2\big)\Big) \tag{37}$$

$$= \frac{1}{Z}\exp\Big(-\beta \sum_{(i,j)\in\mathcal{E}} \|\mathbf{P}_{ij} - \mathbf{X}_i\mathbf{X}_j^\top\|_{\mathrm{F}}^2\Big) \tag{38}$$

where we used the fact that $\|\mathbf{P}_{ij}\|_{\mathrm{F}}^2 = n$ in Equation 37 since $\mathbf{P}_{ij} \in \mathcal{P}_n$ and $Z = C\exp(-\beta n|\mathcal{E}|)$.

In the rest of the proof, we will characterize the normalizing constant $Z_{ij} = \int_{\mathcal{P}_n} \exp(2\beta \mathrm{tr}(\mathbf{P}_{ij}^\top \mathbf{X}_{ij})\mathrm{d}\mathbf{P}_{ij}$. Here, we denote the the counting measure on $\mathcal{P}_n$ as $\mathrm{d}\mathbf{P}_{ij}$.

We start by decomposing $\mathbf{X}_{ij}$ via Birkhoff-von Neumann theorem:

$$\mathbf{X}_{ij} = \sum_{b=1}^{B_{ij}} \theta_{ij,b}\mathbf{M}_{ij,b}, \quad \sum_{b=1}^{B_{ij}} \theta_{ij,b} = 1, \quad B_{ij} \in \mathbb{Z}_+, \quad \theta_{ij,b} \geq 0,\ \mathbf{M}_{ij,b} \in \mathcal{P}_n, \quad \forall b = 1, \ldots, B_{ij}. \tag{39}$$

Then, we have:

$$\begin{aligned}
Z_{ij} &= \int_{\mathcal{P}_n} \exp\Big(2\beta \mathrm{tr}(\mathbf{P}_{ij}^\top \mathbf{X}_{ij}\Big)\mathrm{d}\mathbf{P}_{ij} && (40)\\
&= \int_{\mathcal{P}_n} \exp\Big(2\beta \mathrm{tr}(\mathbf{P}_{ij}^\top \sum_{b=1}^{B_{ij}} \theta_{ij,b}\mathbf{M}_{ij,b}\Big)\mathrm{d}\mathbf{P}_{ij} && (41)\\
&= \int_{\mathcal{P}_n} \exp\Big(2\beta \sum_{b=1}^{B_{ij}} \theta_{ij,b}\mathrm{tr}(\mathbf{P}_{ij}^\top \mathbf{M}_{ij,b})\Big)\mathrm{d}\mathbf{P}_{ij} && (42)\\
&\geq \exp\Big(\int_{\mathcal{P}_n} 2\beta \sum_{b=1}^{B_{ij}} \theta_{ij,b}\mathrm{tr}(\mathbf{P}_{ij}^\top \mathbf{M}_{ij,b})\mathrm{d}\mathbf{P}_{ij}\Big) && (43)\\
&= \exp\Big(\sum_{b=1}^{B_{ij}} \int_{\mathcal{P}_n} 2\beta\theta_{ij,b}\mathrm{tr}(\mathbf{P}_{ij}^\top \mathbf{M}_{ij,b})\mathrm{d}\mathbf{P}_{ij}\Big) && (44)
\end{aligned}$$

where we used Jensen's inequality in (43). Here, we observe that $\int_{\mathcal{P}_n} 2\beta\theta_{ij,b}\mathrm{tr}(\mathbf{P}_{ij}^\top \mathbf{M}_{ij,b})\mathrm{d}\mathbf{P}_{ij}$ is similar to the normalization constant of a Mallows model [21], and it only depends on β and $\theta_{ij,b}$. Hence, we conclude that

$$\begin{aligned}
Z_{ij} &\geq \prod_{b=1}^{B_{ij}} \exp\Big(\int_{\mathcal{P}_n} 2\beta\theta_{ij,b}\mathrm{tr}(\mathbf{P}_{ij}^\top \mathbf{M}_{ij,b})\mathrm{d}\mathbf{P}_{ij}\Big) && (45)\\
&:= \prod_{b=1}^{B_{ij}} f(\beta, \theta_{ij,b}) && (46)
\end{aligned}$$

where f is an increasing function of $\beta, \theta_{ij,b}$ since $\mathrm{tr}(\mathbf{P}_{ij}^\top \mathbf{M}_{ij,b}) \geq 0$. This completes the proof. □

D. Derivation of the Retraction Euler Integrator

We start by recalling the SDE

$$\mathrm{d}\tilde{\mathbf{X}}_t = (-\mathbf{G}^{-1}\nabla_{\tilde{\mathbf{X}}} U_\lambda(\tilde{\mathbf{X}}_t) + \mathbf{\Gamma}_t)\mathrm{d}t + \sqrt{2/\beta\mathbf{G}^{-1}}\mathrm{dB}_t. \tag{47}$$

By [91], we know that

$$\mathbf{\Gamma}_t = \frac{1}{2}\mathbf{G}^{-1}\nabla_{\tilde{\mathbf{X}}} \log|\mathbf{G}|. \tag{48}$$

By using this identity in Equation 47, we obtain:

$$\mathrm{d}\tilde{\mathbf{X}}_t = -\mathbf{G}^{-1}(\nabla_{\tilde{\mathbf{X}}} U_\lambda(\tilde{\mathbf{X}}_t) + \frac{1}{2}\nabla_{\tilde{\mathbf{X}}} \log|\mathbf{G}|)\mathrm{d}t + \sqrt{2/\beta\mathbf{G}^{-1}}\mathrm{dB}_t. \tag{49}$$

By using a similar notation to [54], we rewrite the above equation as follows:

$$\mathrm{d}\tilde{\mathbf{X}}_t = -\mathbf{G}^{-1}(\nabla_{\tilde{\mathbf{X}}} U_\lambda(\tilde{\mathbf{X}}_t) + \frac{1}{2}\nabla_{\tilde{\mathbf{X}}} \log|\mathbf{G}|)\mathrm{d}t + \mathbf{G}^{-1}\mathbf{M}^\top \mathcal{N}(0, 2\beta\mathbf{I}) \tag{50}$$

where $\mathcal{N}$ denotes the Gaussian distribution and $[\mathbf{M}]_{ij} = \partial[\mathbf{X}]_{ij}/\partial[\tilde{\mathbf{X}}]_{ij}$. We multiply each side of the above equation by $\mathbf{M}$ and use the property $\nabla_{\tilde{\mathbf{X}}} = \mathbf{M}^\top \nabla_{\mathbf{X}}$ [54], which yields:

$$\mathrm{d}\mathbf{X}_t = -\mathbf{M}\mathbf{G}^{-1}\mathbf{M}^\top \nabla_{\mathbf{X}} U(\mathbf{X}_t)\mathrm{d}t + \mathbf{M}\mathbf{G}^{-1}\mathbf{M}^\top \mathcal{N}(0, 2\beta\mathbf{I}) \tag{51}$$

$$= \mathbf{M}(\mathbf{M}^\top\mathbf{M})^{-1}\mathbf{M}^\top \Big(-\nabla_{\mathbf{X}} U(\mathbf{X}_t)\mathrm{d}t + \mathcal{N}(0, 2\beta\mathbf{I})\Big). \tag{52}$$

Here we used the area formula (Equation 11 in the main paper) and the fact that $\mathbf{G} = \mathbf{M}^\top\mathbf{M}$.

The term $\mathbf{M}(\mathbf{M}^\top\mathbf{M})^{-1}\mathbf{M}^\top$ turns out the be the projection operator to the tangent space of $\mathbf{X}$ [15]. Therefore, the usual geodesic integrator would consist of the following steps at iteration k:

- Set a small step size h
- Compute the term $-h\nabla_{\mathbf{X}} U(\mathbf{X}_t) + \sqrt{2h/\beta}\mathbf{Z}$, with $\mathbf{Z} \sim \mathcal{N}(0, \mathbf{I})$
- Obtain the 'direction' by projecting the result of the previous step on the tangent space
- Move the current iterate on the geodesic determined by the direction that was obtained in the previous step.

Unfortunately, the last step of the integrator above cannot be computed in the case of $\mathcal{DP}_n$. Therefore, we replace it with moving the current iterate by using the retraction operator, which yields the update equations given in the main paper.

E. Details on the Synthetic Dataset

We now give further details on the synthetic data used in the paper. We synthesize our data by displacing a 4×4 2D grid to simulate 5 different images and scrambling the order of correspondence. Shown in Fig. 7a, the ground truth corresponds to an identity ordering where the i^{th} element of each grid is mapped to i^{th} node of the other. Note that the graph is fully connected but we show only consecutive connections. To simulate noisy data, we introduce 16 random swaps into the ground truth as shown in Fig. 7b. We also randomize the initialization in a similar way. On this data we first run a baseline method where instead of restricting ourselves to the Birkhoff Polytope, we use the paths of the orthogonal group $O(N)$. Our second baseline is the prosperous method of Maset *et al*. [59]. Note that, regardless of the initialization (Fig 7e,f) our method is capable of arriving at visually more satisfactory local minimum. This validates that for complex problems such as the one at hand, respecting the geometry of the constraints is crucial. Our algorithm is successful at that and hence is able to find high quality solutions. In the figure the more *parallel* the lines are, the better, depicting closeness to the ground truth.

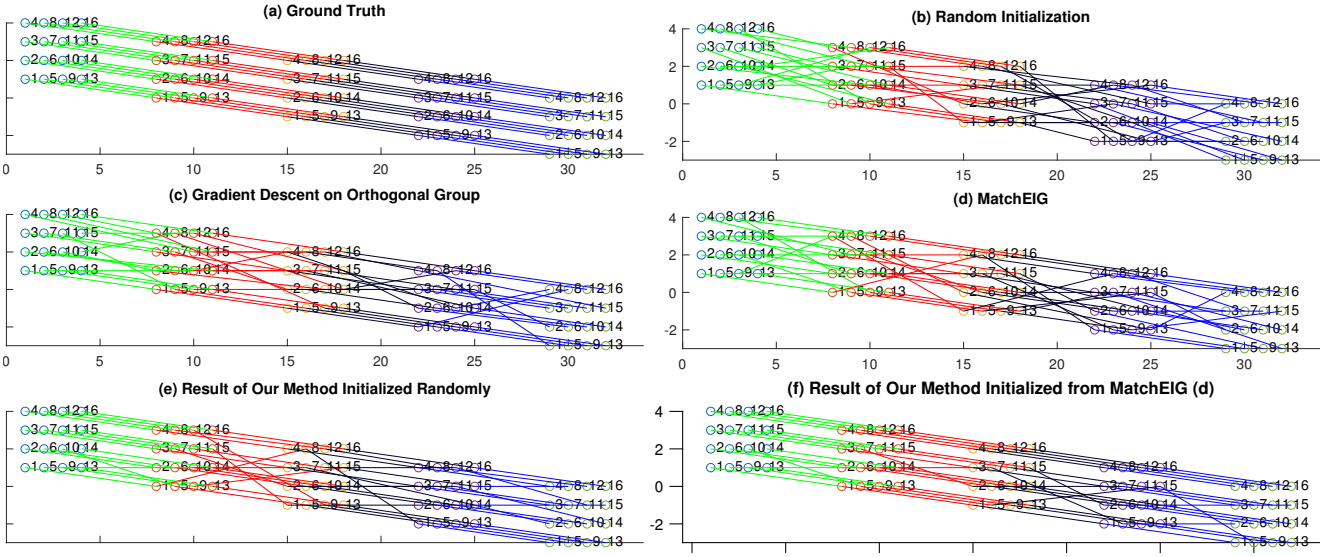


Figure 7. Results from running our method on the synthetic dataset as well as some baseline methods such as Maset *et al*. [59] and minimizing our energy on the orthogonal group. Note that while using the same energy function, considering the Birkhoff Polytope (ours) rather than the orthogonal group leads to much more visually appealing results with higher recall.

F. Examples on the Willow Dataset

Finding the Optimum Solution We now show matches visualized after running our synchronization on the Willow dataset [20]. For the sake of space, we have omitted some of those results from the paper. Fig 8 plots our findings, where

our algorithm is able to converge from challenging initial matches. Note that these results only show our the MAP estimates explained in the main paper. It is possible to extend our approach with some form of geometric constraints as in Wang *et al*. [88]. We leave this as a future work.

Uncertainty Estimation There are not many well accepted standards in evaluating the uncertainty. Hence in the main paper, we resorted what we refer as the *top-K* error. We will now briefly supply more details on this. The purpose of the experiment is to measure the quality of the sample proposals generated by our Birkhoff-RLMC. To this end, we introduce a random sampler that generates, K arbitrary solutions that are highly likely to be irrelevant (bad proposals). These solutions alone do not solve the problem. However if we were to consider the result correct whenever either the found optimum or the random proposal contains the correct match i.e. append the additional $K-1$ samples to the solution set, then, even with a random sampler we are guaranteed to increase the recall. Similarly, samples from Birkhoff-RLMC will also yield higher recall. The question then is: Can Birkhoff-RLMC do better than a random sampler? To give an answer, we basically record the relative improvement in recall both for the random sampler and Birkhoff-RLMC. The *top-K error* for different choices of K is what we presented in Table 2 of the main paper. We further visualize these solutions in the columns (c) to (f) of the same table. To do so, we simply retain the assignments with the K-highest scores in the solutions $\mathbf{X}$. Note that the entire $\mathbf{X}$ acts as a confidence map itself (Column d). We then use the MATLAB command *imagesc* on the $\mathbf{X}$ with the *jet* colormap.

Next, we provide insights into how the sampler works in practice. Fig. 9 plots the objective value attained as the iterations of the Birkhoff-RLMC sampler proceeds. For different values of β these plots look different. Here we use $\beta = 0.08$ and show both on Motorbike and Winebottle samples (used above) and for 1000 iterations, the behaviour of sample selection. In the paper, we have accumulated these samples and estimated the confidence maps. Note that occasionally, the sampler can discover better solutions than the one being provided. This is due to two reasons: 1) rarely, we could jump over local minima, 2) the initial solution is a discrete one and it is often plausible to have a doubly stochastic (relaxed) solution matrices that have lower cost.

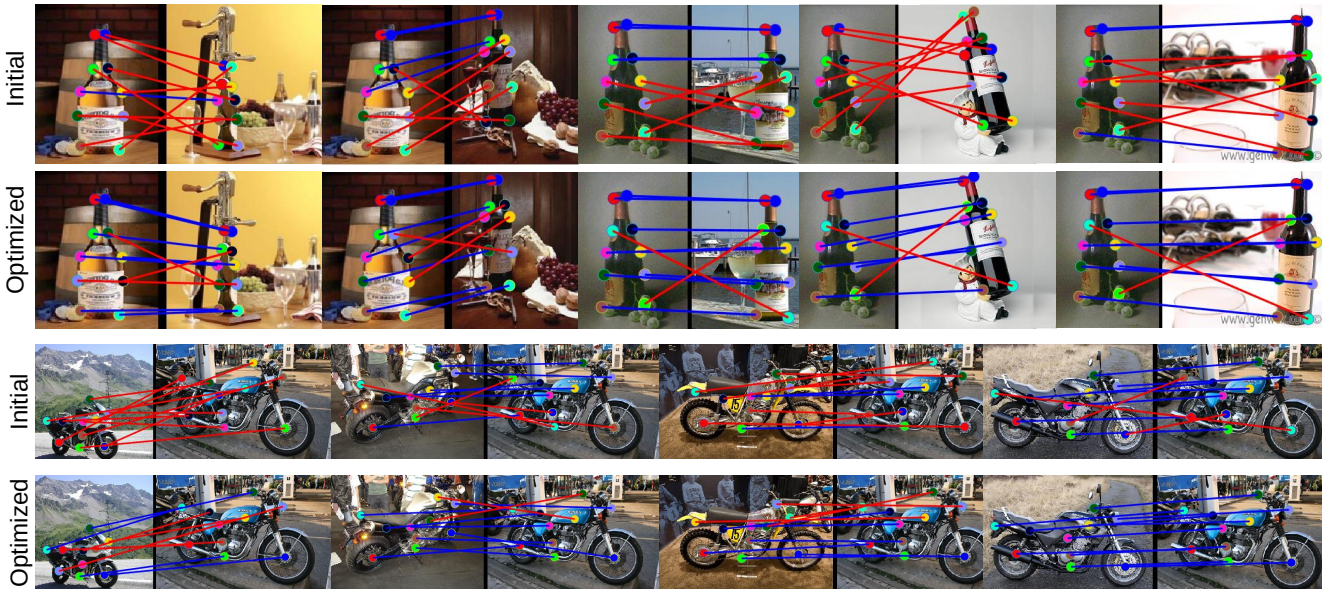
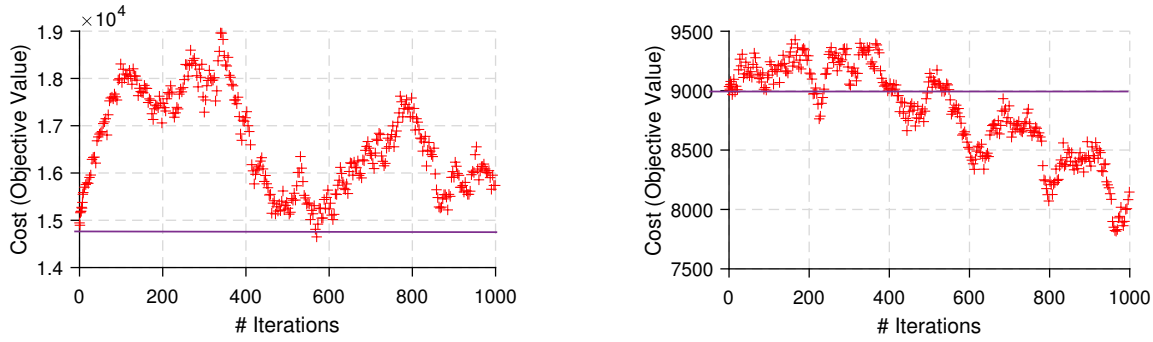


Figure 8. Matching results on sample images from the *Winebottle* (**top-2-rows**) and *Motorbike* (**bottom-2-rows**) datasets in the form of correspondences. The first sub-row of each row shows the initial matches computed via running a pairwise Hungarian algorithm, whereas the second row (tagged *Optimized*) shows our solution. Colored dots are the corresponding points. A line segment is shown in red if it is found to be wrong when checked against the ground truth. Likewise, a blue indicates a correct match. For both of the datasets we use the joint information by optimizing for the cycle consistency, whereas the pairwise solutions make no use of such multiview correspondences. Note that thanks to the cycle consistency, once a match is correctly identified, its associated point has the tendency to be correctly matched across the entire dataset.

G. Application to Establishing 3D Correspondences

We now apply our algorithm to solve correspondence problems on isolated 3D shapes provided by the synthetic Tosca dataset [13] that is composed of non-rigidly deforming 3D meshes of various objects such as cat, human, Centaur, dog, wolf



(a) Sampling on the Winebottle

(b) Sampling on the Motorbike

Figure 9. Iterates of our sampling algorithm. With $\beta = 0.085$, our Birkhoff-RLMC sampler wanders around the local mode and draws samples on the Birkhoff Polytope. It can also happen that better solutions are found and returned. Purple line indicates the starting point, that is only a slight perturbation of the found solution.

and horse. The *cat* object from this dataset along with ten of its selected ground truth correspondences is shown in Fig. 10. This is of great concern among the 3D vision community and the availability of the complete shapes plays well with the assumptions of our algorithm, i.e. permutations are *total*. It is important to mention that when initial correspondences are good ($\sim 80\%$) all methods, including ours can obtain 100% accuracy [42]. Therefore, to be able to put our method to test, we will intentionally degrade the initial correspondence estimation algorithm we use.

Initialization Analogous to the 2D scenario, we obtain the initial correspondences by running a siamese Point-Net [69] like network regressing the assignments between two shapes. Unlike 2D, we do not need to take care of occlusions, visibility or missing keypoint locations as 3D shapes can be full and clean as in this dataset. On the average, we split half of the datasets for training and the other half for testing. Note that such amount of data is really insufficient to train this network, resulting in suboptimal predictions. We gradually downsample the point sets uniformly to sizes of $N_s = \{200, 50, 20\}$ to get the keypoints. At this point, it is possible to use sophisticated keypoint prediction strategies to establish the optimum cardinality and the set of points under correspondence. Note that it is sufficient to compute the keypoints per shape as we do not assume the presence of a particular order. Each keypoint is matched to another by the network prediction. The final stage of the prediction results in a soft assignment matrix on which we apply Hungarian algorithm to give rise to initial matches.

Figure 10. Visualizations of the *cat* object from the Tosca dataset with the ground truth correspondences depicted.

Preliminary Results We run, on these initial sets of correspondences, our algorithm and the state of the art methods as described in the main paper. We count and report the percentage of correctly matched correspondences (recall) in Tab. 3 for only two objects, *Cat* and *Michael*. Running on the full dataset is a future work. It is seen that our algorithm consistently outperforms the competing approaches. The difference is less visible when the number of samples start dominating the number of edges. This is because none of the algorithms are able to find enough consistency votes to correct for the errors.

Table 3. Correspondence refinement on 3D inputs. We show our results on the meshes of Tosca [13] dataset. The cells show the percentage of correctly detected matches (*recall*).

	N=200				N=50				N=20			
	Initial	MatchEIG	Wang et al.	Ours	Initial	MatchEIG	Wang et al.	Ours	Initial	MatchEIG	Wang et al.	Ours
Cat	38.42%	32.92%	37.83%	**39.08%**	53.38%	55.42%	56.13%	**56.93%**	35.56%	37.33%	38.33%	**40.33%**
Michael	30.64%	34.39%	35.92%	**36.12%**	46.40%	52.23%	**54.93%**	54.62%	45.76%	51.05%	51.63%	**55.95%**