IEEE Open Journal of Signal Processing

Received 3 May 2023; revised 19 August 2023; accepted 29 August 2023. Date of publication 1 September 2023; date of current version 14 September 2023. The review of this article was arranged by Associate Editor Vincenzo Matta.

Digital Object Identifier 10.1109/OJSP.2023.3311354

Unrolled Algorithms for Group Synchronization

NOAM JANCO AND TAMIR BENDORY (Senior Member, IEEE)

School of Electrical Engineering, Tel Aviv University, Tel Aviv 69978, Israel

CORRESPONDING AUTHOR: TAMIR BENDORY (e-mail: noamjanco@gmail.com)

This work was supported in part by the NSF-BSF under Grant 2019752, in part by BSF under Grant 2020159, and in part by ISF under Grant 1924/21.

ABSTRACT The group synchronization problem involves estimating a collection of group elements from noisy measurements of their pairwise ratios. This task is a key component in many computational problems, including the molecular reconstruction problem in single-particle cryo-electron microscopy (cryo-EM). The standard methods to estimate the group elements are based on iteratively applying linear and non-linear operators, and are not necessarily optimal. Motivated by the structural similarity to deep neural networks, we adopt the concept of algorithm unrolling, where training data is used to optimize the algorithm. We design unrolled algorithms for several group synchronization instances, including synchronization over the group of 3-D rotations: the synchronization problem in cryo-EM. We also apply a similar approach to the multi-reference alignment problem. We show by numerical experiments that the unrolling strategy outperforms existing synchronization algorithms in a wide variety of scenarios.

INDEX TERMS Algorithm unrolling, group synchronization, multi-reference alignment.

I. INTRODUCTION

Given a group G, the group synchronization problem entails estimating N elements $g_1, \ldots, g_N \in G$ from their noisy pairwise ratios $g_{ij} \approx g_i g_j^{-1}$. Since $g_i g_j^{-1} = (g_i g)(g_j g)^{-1}$ for any $g \in G$, the group elements can be estimated up to a right multiplication by some $g \in G$. A canonical example is the angular synchronization problem of estimating N angles $\theta_1, \ldots, \theta_N \in [0, 2\pi)$ from their noisy offsets $\theta_{ij} \approx (\theta_i - \theta_j) \bmod 2\pi$; this problem corresponds to synchronization over the group of complex numbers on the unit circle $U(1)$ [1], [2], [3], [4].

Under the standard additive Gaussian noise model, the maximum likelihood estimator (MLE) of the angular synchronization problem can be formulated as the solution of a non-convex optimization problem on the manifold of product of circles:

$$\max_{z \in \mathbb{C}_1^N} z^* H z, \tag{I.1}$$

where $H_{ij} = e^{\iota \theta_{ij}}$ is the measurement matrix, $\iota = \sqrt{-1}$, and $\mathbb{C}_1^N := \{z \in \mathbb{C}^N : |z_1| = \ldots = |z_N| = 1\}$. Singer [1] proposed to solve (I.1) by extracting the leading eigenvector of H using the power method: given an initial estimate of the sought angles, the power method iteratively applies the matrix H to the current estimate and then normalizes its norm. In follow-up papers, Boumal [2] suggested an alternative normalization strategy, and Perry et al. [5] developed an algorithm which is inspired by the approximate message passing (AMP) framework. These strategies can be naturally extended to additional group synchronization setups. We describe all these methods in detail in Section II. For our purposes, it is important to note that the t-th iteration of all these methods follows the same structure:

$$z^{(t)} = f(H, z^{(t-1)}, z^{(t-2)}), \tag{I.2}$$

for some non-linear function f. Specifically, at each iteration, the current estimate is acted upon by a linear operator, followed by a non-linear function. This structural resemblance to the blueprint of a neural network layer is the cornerstone of this work.

The group synchronization problem is an important component in a variety of scientific, engineering, and mathematical problems, including the structure from motion problem [6], sensor network localization [7], phase retrieval [8], [9], [10], ranking [11], community detection [12], and synchronization of the rigid motion group [13], [14], [15], [16], the dihedral group [17], and the permutation group [18].

This work is mainly motivated by the problem of 3-D molecular structure reconstruction using cryo-EM [19], in which each observation is a noisy tomographic projection of the molecular structure, taken from some unknown viewing direction. One approach to solving the cryo-EM problem is to estimate the missing 3-D rotations from the observations and then recover the 3-D structure as a linear problem. This methodology is used to constitute ab initio models [20]. In [21], [22], [23], [24], it was shown that the pairwise relative rotations can be estimated from the observations based on the common lines property. Therefore, the cryo-EM reconstruction problem boils down to a synchronization problem over the group of 3-D rotations SO(3).

Motivated by the fact that existing synchronization methods are not optimal, and the resemblance of the iteration (I.2) to the general structure of a modern neural network layer, we adopt the approach of algorithm unrolling [25], to develop an efficient, interpretable neural network that outperforms existing methods. The underlying idea of algorithm unrolling, first introduced in the seminal work of Gregor and LeCun [26], is to exploit existing iterative algorithms and optimize them using training data. Specifically, each iteration of the algorithm is represented as a layer of a network, and concatenating these layers forms a deep neural network. Passing through the network is analogous to executing the iterative algorithm for a fixed number of steps. The network can be trained using back-propagation, resulting in model parameters that are learned from training samples. Thus, the trained network can be naturally interpreted as an optimized algorithm. This is especially important since, while the past decade has witnessed the unprecedented success of deep learning techniques in numerous applications, most deep learning techniques are purely data-driven, and the underlying structures are hard to interpret. The unrolled networks are parameter efficient, require less training data, and are less susceptible to overfitting. Moreover, the unrolled networks naturally inherit prior structures and domain knowledge, leading to better generalization. The algorithm unrolling approach has been adopted for various tasks in recent years, including compressive sensing [27], image processing [28], [29], [30], [31], [32], graph signal processing [33], biological imaging [34], to name but a few. We refer the readers to a recent survey on algorithm unrolling and references therein [25]. Fig. 1 demonstrates the concept of algorithm unrolling for the synchronization problem over the group $\mathbb{Z}/2$; see Section II.

Existing synchronization algorithms assume a certain model of the measurements and use an approximation to solve the non-convex optimization problem, which is not necessarily optimal. For example, the noise is typically not i.i.d. Gaussian. In contrast, the unrolling algorithm, which is data-driven, with inductive bias derived from the problem structure, has the ability to learn a better model, which empirically leads to improved error and improved efficiency. The downside of using a learned algorithm is the complexity of the solution, which requires training on labeled data, and the fact that it might suffer from overfitting. It is known, as we also observed, that the unrolling technique mitigates such issues, compared to a fully data-driven approach.

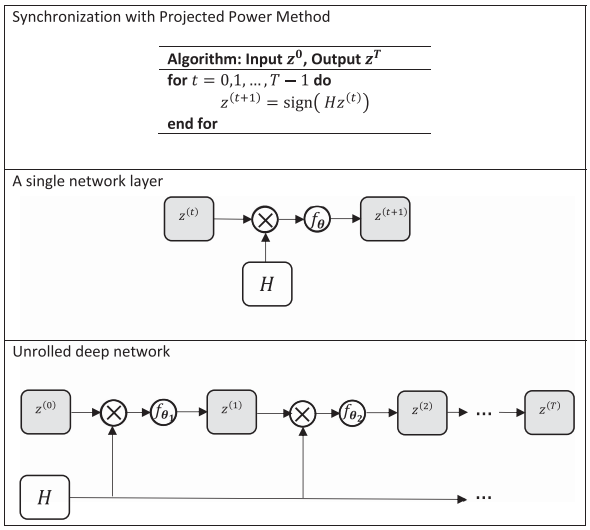


FIGURE 1. Concept of an unrolled algorithm for $\mathbb{Z}/2$ synchronization based on the projected power method. The upper panel shows the standard projected power method for $\mathbb{Z}/2$ synchronization with T iterations; see Section II-A2. The middle panel illustrates a single iteration of the algorithm in the form of a single-layer network, where the sign operator is replaced by a general, learned non-linear function f_θ. The bottom panel shows the concatenation of T layers into an unrolled deep network. Each layer may have a separate set of parameters. The Onsager correction term that uses $z^{(t-1)}$ is omitted for simplicity.

We also study the application of the unrolling approach to the multi-reference alignment (MRA) problem. MRA is the problem of estimating a signal from its multiple noisy copies, each acted upon by a random group element. The computational and statistical properties of the MRA problem have been analyzed thoroughly in recent years; see [35], [36], [37], [38], [39], [40], [41], [42], [43], [44], [45], [46]. Group synchronization is often used to solve the MRA problem in high SNR regimes, by first estimating the pairwise ratios between the group elements from the noisy observations, and then estimating the group elements themselves as a synchronization problem. Given an accurate estimate of the random group elements, the MRA problem reduces to a linear inverse problem, which is much easier to solve. Importantly, in contrast to group synchronization, the goal in MRA is to estimate the underlying signal, while the group elements are nuisance variables whose estimation is merely an intermediate step.

The rest of the article is organized as follows. In Section II we introduce three particular cases of group synchronization and two MRA models, and present existing methods to solve them. Section III introduces the proposed unrolled algorithms, and Section IV shows numerical results. Finally, Section V concludes the article, and discusses future work.

II. GROUP SYNCHRONIZATION, MULTI-REFERENCE ALIGNMENT, AND EXISTING SOLUTIONS

In this section, we introduce three group synchronization and two MRA models. We also elaborate on three different

methods to estimate group elements. These methods are the keystone of the unrolled algorithms described in the next section.

A. $\mathbb{Z}/2$ SYNCHRONIZATION

We begin with the simplest group synchronization problem over the group $\mathbb{Z}/2$. The goal is to estimate a signal $z \in \{\pm 1\}^N$ from the noisy measurement matrix:

$$H = \frac{\lambda}{N} z z^T + \frac{1}{\sqrt{N}} W, \tag{II.1}$$

where $W_{ij} = W_{ji} \sim \mathcal{N}(0, 1)$, and $\lambda > 0$ is a signal-to-noise ratio (SNR) parameter. The scaling is such that the signal and noise components of the observed data are of comparable magnitudes. The diagonal entries of W follow the same distribution. We also assume that each entry of z is drawn i.i.d. from a uniform distribution over ± 1. We can only hope to estimate z up to a sign, due to the symmetry of the problem.

The $\mathbb{Z}/2$ synchronization problem is associated with the maximum likelihood estimation problem:

$$\max_{z \in \mathbb{R}_1^N} z^T H z, \tag{II.2}$$

where $\mathbb{R}_1^N := \{z \in \mathbb{R}^N : |z_1| = \ldots = |z_N| = 1\}$. This is a nonconvex optimization problem. We now describe different existing iterative algorithms to solve (II.2). All algorithms are initialized with small random values in $[-1, 1]$. Specifically, in our numerical experiments, the algorithms are initialized by $z^{(0)}, z^{(-1)} \sim \mathcal{N}(0, 10^{-2} I)$.

1) POWER METHOD (PM)

In [1], Singer proposed a spectral approach (in the context of $U(1)$ synchronization) that relaxes (II.2) to

$$\max_{z \in \mathbb{R}^N, \|z\|^2 = N} z^T H z = \max_{z \in \mathbb{R}^N, \|z\|^2 = N} N \frac{z^T H z}{\|z\|^2}. \tag{II.3}$$

The expression in (II.3) is known as the Rayleigh quotient and is maximized by the leading eigenvector of H that corresponds to the largest eigenvalue. This eigenvector can be computed using the power method, whose $(t+1)$-th iteration reads:

$$z^{(t+1)} = \frac{H z^{(t)}}{\|H z^{(t)}\|}. \tag{II.4}$$

After the last iteration T, the output is projected onto the $\mathbb{Z}/2$ group by $z(T) = \text{sign}(z(T))$, where sign() is the sign function, acting separately on each entry of the vector.

2) PROJECTED POWER METHOD (PPM)

The projected power method [2] suggests replacing the global normalization (II.4) by an entrywise projection onto the group. Specifically, the $(t+1)$-th iteration reads:

$$z^{(t+1)} = \text{sign}(H z^{(t)}). \tag{II.5}$$

3) APPROXIMATE MESSAGE PASSING (AMP)

Perry et al. [5] proposed an algorithm that is inspired by the AMP framework. For the $\mathbb{Z}/2$ synchronization, its $(t+1)$-th iteration reads:

$$z^{(t+1)} = \tanh(c^{(t+1)}), \tag{II.6}$$

where

$$c^{(t+1)} = \lambda H z^{(t)} - \lambda^2 (1 - \langle (z^{(t)})^2 \rangle) z^{(t-1)}, \tag{II.7}$$

and $\langle \cdot \rangle$ denotes averaging over the vector entries. The second term in (II.7) is called the Onsager correction term and is related to backtracking messages in graphical model [5]. In the setting of $\mathbb{Z}/2$ synchronization, an algorithm equivalent to the AMP [5] was analyzed in [47], where a statistical optimality property is proven: if AMP is warm-started with a state v_0 with nontrivial correlation with the truth, then it converges to an estimate of x that achieves minimum mean-squared error (MMSE) asymptotically as $n \to \infty$. In [5], they conjecture based on ideas from statistical physics that in many regimes besides $\mathbb{Z}/2$ synchronization, the AMP algorithm is statistically optimal, as the matrix dimensions approaches infinity.

We underscore that all the methods mentioned above share a similar structure: the current estimate of the group elements is multiplied by the measurement matrix, followed by a nonlinear operation.

B. $U(1)$ SYNCHRONIZATION

Next, we consider the synchronization problem over the group $U(1)$ of complex numbers with unit modulus. The formulation is similar to Section II-A, but the goal is to estimate $z \in \mathbb{C}_1^N$. In this case, W is a Hermitian matrix whose entries are distributed independently (up to symmetry) according to the standard complex normal distribution $\mathcal{CN}(0, 1)$. We assume that each entry of z is drawn i.i.d. from a uniform distribution on the unit circle. Due to symmetry considerations, we can only hope to estimate z up to a global element of $U(1)$. We describe different existing iterative algorithms to solve the optimization problem. All algorithms are initialized with small random values. In our experiments, the algorithms are initialized with $z^{(0)}, z^{(-1)} \sim \mathcal{CN}(0, 2 \cdot 10^{-4} I)$

1) POWER METHOD (PM)

Using a relaxation similar to (II.3) with $z \in \mathbb{C}^N$ instead of $z \in \mathbb{R}^N$, we get power iterations as in (II.4).

2) PROJECTED POWER METHOD (PPM)

Similarly to (II.5), the $(t+1)$-th iteration of the PPM reads:

$$z^{(t+1)} = \text{phase}(H z^{(t)}), \tag{II.8}$$

where $\text{phase}(z)[i] = z[i]/|z[i]|$.

3) APPROXIMATE MESSAGE PASSING (AMP)

Following [5], for each $i = 1, \ldots, N$, the $(t+1)$-th iteration of the AMP algorithm reads:

$$z^{(t+1)}[i] = f\left(|c^{(t+1)}[i]|\right) \frac{c^{(t+1)}[i]}{|c^{(t+1)}[i]|}, \tag{II.9}$$

where $f(t) = I_1(2t)/I_0(2t)$, I_k denotes the modified Bessel functions of the first kind of order k, and

$$c^{(t+1)} = \lambda H z^{(t)} - \lambda^2 (1 - \langle |z^{(t)}|^2 \rangle) z^{(t-1)}. \tag{II.10}$$

C. SO(3) SYNCHRONIZATION

$SO(3)$ is the group of 3-D rotations. Each element of $SO(3)$ can be represented by a 3×3 matrix R_i that satisfies $\det(R_i) = 1$, and $R_i R_i^T = R_i^T R_i = I$, where I is the identity matrix. The $SO(3)$ synchronization problem is to estimate the block matrix

$$R = [R_1^T, \ldots, R_N^T]^T \in \mathbb{R}^{3N \times 3}, \tag{II.11}$$

given the noisy pairwise ratios:

$$H = \frac{\lambda}{N} R R^T + \frac{1}{\sqrt{3N}} W, \tag{II.12}$$

where W is a symmetric matrix whose entries are distributed independently (up to symmetry) as $\mathcal{N}(0, 1)$, and $\lambda > 0$ denotes the SNR parameter. The problem can be associated with the maximum likelihood estimation problem [23]:

$$\max_R R^T H R, \tag{II.13}$$

where $R \in \mathbb{R}^{3N \times 3}$ is of the form (II.11), and each 3×3 block R_i is in $SO(3)$.

1) SPECTRAL METHOD

Similarly to synchronization over $\mathbb{Z}/2$ and $U(1)$, we begin by computing the three leading eigenvectors of $H \in \mathbb{R}^{3N \times 3N}$, which we denote by $\hat{R}_1, \hat{R}_2, \hat{R}_3$. This method is typically called the spectral method [23], and we omit the details of the power iterations for simplicity. Then, we form a matrix $\hat{R} = [\hat{R}_1, \hat{R}_2, \hat{R}_3] \in \mathbb{R}^{3N \times 3}$, and finally each 3×3 block of $\hat{R}$ is projected onto the nearest orthogonal matrix. This projection, denoted by $\text{project}_{SO(3)}$, takes a 3×3 matrix M, computes its SVD factorization $M = U \Sigma V^T$ and replaces the diagonal matrix Σ by an identity matrix so that $\text{project}_{SO(3)}(M) = \pm U V^T$. The sign is chosen so that the determinant is one.

2) PROJECTED POWER METHOD (PPM)

The $(t+1)$-th iteration of the PPM reads:

$$R^{(t+1)} = \text{project}_{SO(3)}(H R^{(t)}). \tag{II.14}$$

To initialize the algorithm, we draw N, 3×3 matrices whose entries are drawn i.i.d. from $\mathcal{N}(0, 1)$, and then project each matrix to the nearest orthogonal matrix as described above.

D. MULTI-REFERENCE ALIGNMENT (MRA)

We consider two MRA setups. In both cases, assuming the SNR is not too low, we first estimate the pairwise ratios between the group elements from the observations. Then, we estimate the group elements using a synchronization algorithm, align the noisy observations, and average out the noise.

1) MRA OVER $\mathbb{Z}/2$

We assume to acquire N measurements of the form

$$y_i = s_i x + \frac{1}{\lambda} \varepsilon_i, \quad i = 1, \ldots, N, \tag{II.15}$$

where $x, \varepsilon_i \in \mathbb{R}^L$, $\varepsilon_i \sim \mathcal{N}(0, I)$ and $s_i \in \{-1, 1\}$. Our goal is to estimate x, up to a sign, from $y_1, \ldots, y_N$, when $s_1, \ldots, s_N$, are unknown.

To estimate x, we first build the pairwise ratio matrix by

$$\frac{H_{ij}}{\|x\|_2^2} = \frac{y_i^T y_j}{\|x\|_2^2} \approx s_i s_j, \tag{II.16}$$

and then estimate the group elements $\{s_i\}_{i=1}^N$ using one of the existing methods for $\mathbb{Z}/2$ synchronization described in Section II-A. Let $\hat{s}_1, \ldots, \hat{s}_N$, be the estimated group elements. Then, the signal can be reconstructed by averaging

$$\hat{x} = \frac{1}{N} \sum_{i=1}^N \hat{s}_i y_i. \tag{II.17}$$

We emphasize that, in contrast to the synchronization problem, the error in (II.16) is not Gaussian anymore; in fact, the error is correlated:

$$H_{ij} = \frac{\lambda}{N} y_i^T y_j = \frac{\lambda}{N} s_i s_j \|x\|_2^2 + w_{i,j}, \tag{II.18}$$

where $w_{i,j} = \frac{1}{N}(x^T(s_j \varepsilon_i + s_i \varepsilon_j) + \frac{1}{\lambda} \varepsilon_i^T \varepsilon_j)$. Note that $\mathbb{E}[w_{i,j} w_{i,k}] = \frac{s_j s_k}{N^2} \mathbb{E}[(x^T \varepsilon_i)^2] \neq 0$, where the expectation is taken with respect to the noise terms.

2) MRA OVER THE GROUP $\mathbb{Z}/L$ OF CIRCULAR SHIFTS

Now, we consider a set of measurements of the form

$$y_i = R_{s_i} x + \frac{1}{\lambda} \varepsilon_i \quad i = 1, \ldots, N, \tag{II.19}$$

where $x \in \mathbb{R}^L$ is sought signal, R_s is a circular shift operator, that is, $R_s(x)[i] = x[(i - s) \bmod L]$, $s \sim U[0, L-1]$, and $\varepsilon_i \sim \mathcal{N}(0, I)$. We wish to estimate x, up to a circular shift, from $y_1, \ldots, y_N$, when $s_1, \ldots, s_N$ are unknown.

To estimate the signal, we first estimate the pairwise ratio between the group elements (namely, the relative circular shift) by taking the maximum of the cross-correlation between pairs of observations. This can be computed efficiently using the FFT algorithm by the relation

$$s_{ij} = \arg\max \mathcal{F}^{-1}(\mathcal{F}(y_i) \circ \mathcal{F}^*(y_j)), \tag{II.20}$$

where $\mathcal{F}$ stands for the Fourier transform and $\circ$ is an element-wise multiplication. Then, we construct the pairwise matrix:

$$H_{ij} = \frac{\lambda}{N} e^{\iota 2\pi \frac{s_{ij}}{L}}, \tag{II.21}$$

and estimate the group elements using one of the existing methods for $U(1)$ synchronization described in Section II-B. We keep the normalization to be consistent with the scaling of the synchronization model when the pairwise ratios are given.

Let $\hat{s}_1, \ldots, \hat{s}_N$, be the estimates of the group elements. The signal can then be estimated by alignment and averaging.

$$\hat{x} = \frac{1}{N}\sum_{i=1}^{N} R_{-\hat{s}_i} y_i. \tag{II.22}$$

Throughout this work, we assume that the SNR is high enough so that the group elements can be estimated with reasonable accuracy. We mention that when the SNR is very low, the group elements cannot be estimated reliably, and thus the strategy described above will fail. We focus on $\lambda > 1$, in which the spectral method and the projected power method are shown to result in a non-trivial solution [5]. In the MRA problem, the effective SNR is $\lambda\|x\|_2 \approx \lambda\sqrt{L}$, since $x \in \mathbb{R}^L \sim \mathcal{N}(0, I)$. Throughout our experiments, we chose λ values such that the SNR is of the order of 1. Several methods were developed to estimate the signal in such low SNR environments without estimating the group elements, see, for instance, [36], [38], [39].

III. UNROLLED ALGORITHMS FOR GROUP SYNCHRONIZATION

Based on the structural similarity between the group synchronization algorithms described in Section II and deep neural networks, we adopt the concept of algorithm unrolling: mapping each iteration of an iterative algorithm into a learned network layer and stacking the layers together to form a deep neural network. Each layer consists of multiplying the current estimate of group elements with the measurement matrix, $Hz^{(t)}$, as in the iteration formula, but replaces the explicit non-linear function with a learned non-linear function. Each layer has the flexibility to incorporate information from the $(t-1)$-th layer. Specifically, the $(t+1)$-th layer receives as an input the measurement matrix H and the previous estimates $z^{(t)}$ and $z^{(t-1)}$, and is parameterized by a set of weights $\theta^{(t)}$:

$$z^{(t+1)} = \ell_{\theta^{(t)}}(z^{(t)}, z^{(t-1)}, H), \tag{III.1}$$

where ℓ denotes the architecture of the layer. The layers can either share weights or have different weights per layer. Fig. 1 illustrates the concept of an unrolled algorithm for $\mathbb{Z}/2$ synchronization.

In order to train the network, we generate data according to the data generative model, including the relative measurement matrix and the ground truth group elements. The network is trained using stochastic gradient descent to minimize a loss function that measures an error metric (up to a group symmetry) over a batch of samples. Thus, given an initial estimate $z^{(0)}$, we get an estimator for the group elements of the form:

$$\hat{z} = F_\Theta(z^{(0)}, H), \tag{III.2}$$

where Θ is the entire set of weights: $\Theta = [\theta^{(0)}, \ldots, \theta^{(T-1)}]$, and F is the deep neural network function.

While we cannot provide theoretical guarantees, we conjecture that the unrolling algorithm outperforms existing algorithms for the following reasons:

1) Existing solutions are not necessarily optimal, and the error guarantees are for asymptotic settings, whereas we examine the finite-dimensional setting.
2) The analysis of previous algorithms assumes that the errors of the relative group ratios are independent. However, usually, the relative group ratios are estimated from the data (e.g., in cryo-EM), and thus this error model does not hold.
3) The starting point of this work was the resemblance of existing iterative synchronization algorithms to the blueprint of neural networks. We chose to use algorithm unrolling, and not a generic neural network architecture, to benefit from its advantages: interpretable structure, which contains domain knowledge and requires less training data.

In the following subsections, we elaborate on specific network architectures, including the loss functions, for the models introduced in Section II.

A. ARCHITECTURE AND LOSS FUNCTION FOR $\mathbb{Z}/2$ SYNCHRONIZATION

Following the AMP iterations in (II.6) and (II.7), the $(t+1)$-th layer receives as input the measurement matrix $H \in \mathbb{R}^{N\times N}$, and the previous layers' estimates $z^{(t)}, z^{(t-1)} \in \mathbb{R}^N$. The output $z^{(t+1)} \in \mathbb{R}^N$ can be described using the following equations:

$$c = \theta_0 \lambda H z^{(t)} - \lambda^2(1 - \langle(\phi_{\boldsymbol{\theta}_2}(z^{(t)}))^2\rangle)z^{(t-1)}, \tag{III.3}$$

and,

$$z^{(t+1)} = f_{\boldsymbol{\theta}_1}(c), \tag{III.4}$$

where f and ϕ are learned functions parameterized by a set of weights $\boldsymbol{\theta}_1$ and $\boldsymbol{\theta}_2$.

We denote by Dense(N) a linear layer with N neurons, whose input is the previous layer's output, BatchNorm() denotes a batch normalization layer, ReLU() is a relu layer, and tanh() is a hyperbolic tangent layer. The learned function has the following structure: Dense(32) $\rightarrow$ BatchNorm() $\rightarrow$ ReLU() $\rightarrow$ Dense(1) $\rightarrow$ BatchNorm() $\rightarrow$ tanh(), such that its outputs are in the range $[-1, 1]$.

Given a batch of M samples, with ground truth and predicted group elements $\{z_m\}_{m=1}^M$ and $\{\hat{z}_m\}_{m=1}^M$, respectively, we use the following loss function to optimize the weights:

$$\mathcal{L}(\Theta) = 1 - \frac{1}{NM}\sum_{m=1}^{M} |z_m^T \hat{z}_m|, \tag{III.5}$$

where $\Theta = [\boldsymbol{\theta}^{(0)}, \ldots, \boldsymbol{\theta}^{(T-1)}]$ is the set of parameters of the network, and $\boldsymbol{\theta}^{(t)} = [\theta_0^{(t)}, \boldsymbol{\theta}_1^{(t)}, \boldsymbol{\theta}_2^{(t)}]$ is the set of parameters per layer. The loss function (III.5) measures the average alignment error, up to a sign, between the predicted and the ground truth group elements. The absolute value function is required due to the sign symmetry.

B. ARCHITECTURE AND LOSS FUNCTION FOR $U(1)$ SYNCHRONIZATION

Based on the AMP iterations (II.10) and (II.9), the $(t+1)$-th layer receives as input $H_r, H_i \in \mathbb{R}^{N\times N}$, the real and imaginary parts of the measurement matrix, respectively, and $z_r^{(t)}, z_i^{(t)}, z_r^{(t-1)}, z_i^{(t-1)} \in \mathbb{R}^N$: the real and imaginary parts of the estimates of the previous layers. The output $z_r^{(t+1)}, z_i^{(t+1)} \in \mathbb{R}^N$ can be described using the following equations:

$$\begin{aligned} c_r &= \theta_0\lambda(H_r z_r^{(t)} - H_i z_i^{(t)}) - \lambda^2(1 - \langle {z_r^{(t)}}^2 + {z_i^{(t)}}^2\rangle) z_r^{(t-1)}, \\ c_i &= \theta_0\lambda(H_r z_i^{(t)} + H_i z_r^{(t)}) - \lambda^2(1 - \langle {z_r^{(t)}}^2 + {z_i^{(t)}}^2\rangle) z_i^{(t-1)} \end{aligned} \tag{III.6}$$

and

$$\begin{aligned} z_r^{(t+1)}[n] &= \frac{c_r[j]}{\max(|c[j]|, \varepsilon)} f_{\boldsymbol{\theta}_1}(|c[j]|), \\ z_i^{(t+1)}[j] &= \frac{c_i[j]}{\max(|c[j]|, \varepsilon)} f_{\boldsymbol{\theta}_1}(|c[j]|), \end{aligned} \tag{III.7}$$

where $|c[j]| = \sqrt{c_r[j]^2 + c_i[j]^2}$, and $\varepsilon = 10^{-12}$ is a small constant that is introduced for numerical stability. The non-linear function f is a learned function parameterized by a set of weights $\boldsymbol{\theta}_1$ with the following structure: Dense(256) $\rightarrow$ ReLU() $\rightarrow$ Dense(1) $\rightarrow$ tanh(), such that its outputs are within $[-1, 1]$.

Let $\{z_{r_m}\}_{m=1}^M$, $\{z_{i_m}\}_{m=1}^M$ and $\{\hat{z}_{r_m}\}_{m=1}^M$, $\{\hat{z}_{i_m}\}_{m=1}^M$ be the real and imaginary parts of the ground truth and the predicted group elements, respectively, of a batch of M samples. We use the following loss function to optimize the weights:

$$\begin{aligned} \mathcal{L}(\Theta) = 1 - \frac{1}{NM}\sum_{m=1}^{M} \big[&(z_{r_m}^T \hat{z}_{r_m} + z_{i_m}^T \hat{z}_{i_m})^2 \\ &+ (z_{r_m}^T \hat{z}_{i_m} - z_{i_m}^T \hat{z}_{r_m})^2 \big]^{1/2}, \end{aligned} \tag{III.8}$$

where $\Theta = [\boldsymbol{\theta}^{(0)}, \ldots, \boldsymbol{\theta}^{(T-1)}]$ is the set of network's parameters, and $\boldsymbol{\theta}^{(t)} = [\theta_0^{(t)}, \boldsymbol{\theta}_1^{(t)}]$. This loss function measures the alignment between the ground truth and predicted group elements, and it is invariant to a global phase shift (the symmetry of the problem).

C. ARCHITECTURE AND LOSS FUNCTION FOR $SO(3)$ SYNCHRONIZATION

The projection operation in (II.14), which consists of SVD factorization, is non-differentiable, and thus gradients cannot be back-propagated through it during the learning process. Therefore, in order to unroll the projected power method into a differentiable neural network, this projection operation should be replaced. To derive a differentiable projection operation, we start with an alternative method that expresses the nearest orthogonal matrix of a matrix A, denoted by Q, explicitly using the matrix square root: $Q = A(A^TA)^{-\frac{1}{2}}$. This method can be combined with the Babylonian method, and a first-order approximation suggests the following iterations after setting $Q_0 = A/\|A\|_{\mathrm{F}}$ [48]:

$$\begin{aligned} N_i &= Q_i^T Q_i \\ P_i &= \frac{1}{2} Q_i N_i \\ Q_{i+1} &= 2Q_i + P_i N_i - 3P_i. \end{aligned} \tag{III.9}$$

Numerical experiments suggest that this recursion typically converges after 4 iterations. We thus use Q_4 as an estimation for the nearest orthogonal matrix of A, through which gradients can be backpropagated.

The unrolled synchronization algorithm for $SO(3)$ is composed of stacked learned synchronization blocks, followed by a projection block as the last layer. Each learned synchronization block takes on the form:

$$R^{(t+1)} = f_{\boldsymbol{\theta}_1}(HR^{(t)}) + \phi_{\boldsymbol{\theta}_2}(R^{(t-1)}). \tag{III.10}$$

The function implementation consists of the following layers: Reshape input $(M, 3N, 3)$ to $(M, N, 9)$ $\rightarrow$ Dense(hidden neurons) $\rightarrow$ BatchNorm() $\rightarrow$ ReLU() $\rightarrow$ Dense(9) $\rightarrow$ BatchNorm() $\rightarrow$ tanh() $\rightarrow$ Reshape into $(M, 3N, 3)$, where M is the batch size. The first layer reshapes the input such that each 3×3 block is flattened into 9 elements, resulting in a shape of $(M, N, 9)$. The following layers apply the same non-linear functions to each 9-element vector and reshape them back into the dimensions of the input. The function f uses 32 hidden neurons and ϕ uses 9 hidden neurons. The function ϕ acts as the Onsager correction term and slightly improves the results.

The implementation of the projection block is as follows:

1) reshape the input $(M, 3N, 3)$ to $(M, N, 3, 3)$;
2) normalize each 3×3 matrix by its Frobenius norm and apply the four iterations of (III.9);
3) reshape the output of the last stage to $(M, 3N, 3)$.

Given a batch of samples of size M, with ground truth and predicted group elements $\{R_m\}_{m=1}^M$ and $\{\hat{R}_m\}_{m=1}^M$, respectively, we use the following loss function to optimize the weights:

$$\mathcal{L}(\Theta) = 1 - \frac{3}{NM}\sum_{m=1}^{M} \|R_m^T \hat{R}_m\|_{\mathrm{F}}^2, \tag{III.11}$$

where $\Theta = \left[\boldsymbol{\theta}^{(0)}, \ldots, \boldsymbol{\theta}^{(T-1)}\right]$ is the set of network's parameters, and $\boldsymbol{\theta}^{(t)} = [\boldsymbol{\theta}_1^{(t)}, \boldsymbol{\theta}_2^{(t)}]$. The suggested loss measures the alignment between the ground truth and the predicted group element matrices and is invariant under a global rotation.

D. MULTI-REFERENCE ALIGNMENT (MRA)

MRA models differ from group synchronization in two important aspects. First, the goal of the MRA problem is not to estimate the group elements, but the signal itself. Second, the pairwise ratios are not directly available and are estimated from the observations. Therefore, the learning phase of MRA models is slightly different from group synchronization, as described below. We draw M signals from some distribution. Then, for each signal, we generate N noisy measurements according to the MRA statistical model and estimate the pairwise ratio between the corresponding group elements. Given

the pairwise ratio matrix, we solve a group synchronization problem and aim to estimate the signal itself, up to a group action. As we will see below, this process suggests different loss functions than the ones used for group synchronization.

1) MRA OVER $\mathbb{Z}/2$

A direct application of the $\mathbb{Z}/2$ architecture described in Section III-A, when the pairwise ratios are estimated from the noisy measurements, only leads to a small improvement, as will be presented in Section IV. Therefore, we suggest to incorporate the measurements themselves in the loss function of the neural network.

Let $Y_m \in \mathbb{R}^{L\times N}$ be the measurement matrix of the m-th signal $x_m \in \mathbb{R}^L$, so that $Y_m[:,n] \in \mathbb{R}^L$ is the n-th observation of the m-th signal. We suggest the following reconstruction loss:

$$\mathcal{L}_R(\Theta) = \frac{1}{LM}\sum_{m=1}^{M} \min_{s\in\{-1,1\}} \left\| x_m - \frac{s}{N}\sum_{n=1}^{N} Y_m[:,n]\hat{z}_m[n] \right\|^2, \tag{III.12}$$

where $\hat{z}_m \in \mathbb{R}^N$ is the predicted group elements of the network described in Section III-A. The loss function depends on the parameters Θ through the group elements $\{\hat{z}_m[n]\}_{n,m=1}^{N,M}$. Note that the reconstruction loss is invariant to the inherent sign symmetry.

2) MRA OVER THE GROUP $\mathbb{Z}/L$ OF CIRCULAR SHIFTS

Similar to the MRA model described above, when the relative shifts were estimated from the MRA measurements, only a minor improvement in signal estimation was achieved using the architecture of $U(1)$ synchronization from Section III-B. Thus, we aim to work with the measurements directly.

It is more convenient to express the loss function in the Fourier domain, where a circular shift is mapped to a complex exponential. Let $\mathcal{X}_m \in \mathbb{C}^L$ be the Fourier transform of the m-th signal, and let $\mathcal{Y}_m \in \mathbb{C}^{L\times N}$ be the corresponding measurement matrix, where $\mathcal{Y}_m[:,n]$ is the Fourier transform of the n-th measurement of the m-th signal. Let $\mathcal{Y}_{r_m}$ and $\mathcal{Y}_{i_m}$ denote the real and imaginary parts of $\mathcal{Y}_m$, and let $\hat{z}_m \in U(1)$ be the estimated rotation using the synchronization algorithm described in Section III-B. Note that $\hat{z}_m$ lies on the unit circle, whereas the circular shifts are discrete. The real and imaginary parts of the aligned data matrix of the m-th sample can be written as:

$$\begin{aligned}\tilde{\mathcal{Y}}_{r_m}[k,n] &= \cos\left(k\angle\hat{z}_m\right)\mathcal{Y}_{r_m}[k,n] - \sin\left(k\angle\hat{z}_m\right)\mathcal{Y}_{i_m}[k,n],\\ \tilde{\mathcal{Y}}_{i_m}[k,n] &= \cos\left(k\angle\hat{z}_m^T\right)\mathcal{Y}_{i_m}[k,n] + \sin\left(k\angle\hat{z}_m^T\right)\mathcal{Y}_{r_m}[k,n],\end{aligned} \tag{III.13}$$

for $k = 0,\ldots,L-1$. The signal is then estimated by averaging:

$$\hat{\mathcal{X}}_m = \frac{1}{N}\sum_{n=1}^{N}\left(\tilde{\mathcal{Y}}_{r_m}[:,n] + \iota\tilde{\mathcal{Y}}_{i_m}[:,n]\right). \tag{III.14}$$

Therefore, we use the following loss function:

$$\mathcal{L}_R(\Theta) = c\sum_{m=1}^{M}\min_{\phi\in\Phi_P}\sum_{k}\left(\mathcal{X}_m[k] - e^{jk\phi}\hat{\mathcal{X}}_m[k]\right)^2, \tag{III.15}$$

where $c = \frac{1}{L^2M}$ and $\Phi_P = \{\frac{2\pi}{LP}, 2\frac{2\pi}{LP},\ldots,2\pi\}$. In the numerical experiments below, we set $P = 10$.

IV. NUMERICAL EXPERIMENTS

The following experiments examine the average error of the unrolled algorithms and the iterative algorithms described in Section II. In all experiments, we set $N = 20$, and the number of test samples is equal to the number of training samples. The code to reproduce all experiments is publicly available at https://github.com/noamjanco/unrolling_synchronization.

A. $\mathbb{Z}/2$ SYNCHRONIZATION

For a vector of ground truth group elements $z \in \{\pm1\}^N$ and a prediction $\hat{z}$, the alignment error is defined as:

$$\text{error}(z,\hat{z}) = 1 - \frac{|z^T\hat{z}|}{N}. \tag{IV.1}$$

We note that $\text{error}(z,\hat{z}) = 0$ for a perfect estimation, where $\hat{z} = \pm z$. In addition, the error is invariant to a global sign, i.e., $\text{error}(z,-\hat{z}) = \text{error}(z,\hat{z})$.

Each observation of length $N = 20$ was generated according to (II.1), where each entry was drawn i.i.d. from a uniform distribution over ±1. The network was trained using a dataset of size $M = 20000$, with 300 epochs, and a learning rate of 10^{-3}, using the Adam optimizer with a batch size of 128.

The alignment error as a function of depth is presented in Fig. 2, for SNR values of $\lambda = 1.2$, $\lambda = 1.5$, and $\lambda = 2$. We compared the performance of the unrolled algorithm against the alternative algorithms described in Section II-A, where the number of iterations is equal to the depth of the network. The results demonstrate that the unrolled synchronization network achieves better error performance, and the performance gap increases as the SNR decreases.

Fig. 3 shows the alignment error as a function of SNR, with a network of a fixed depth of 9, while the alternative algorithms used 100 iterations. We see that the neural network outperforms the alternative methods in terms of alignment error with much fewer iterations.

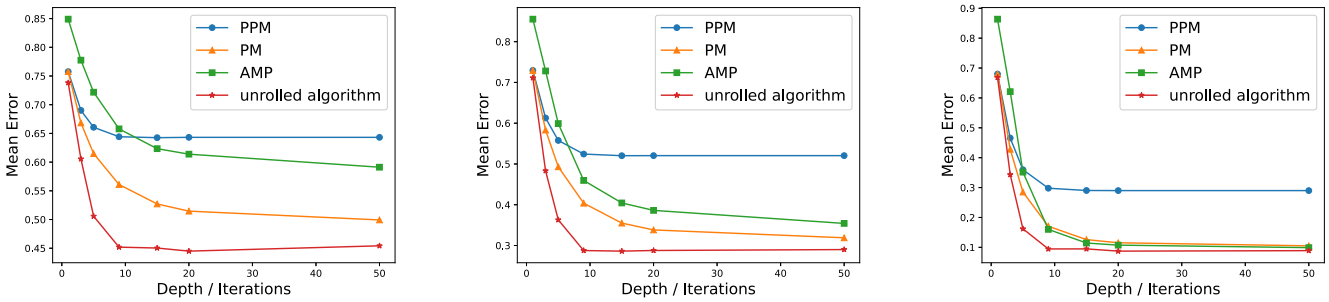
B. $U(1)$ SYNCHRONIZATION

We define the error between the vector of the ground truth group elements $z \in \mathbb{C}_1^N$ and a prediction $\hat{z} \in \mathbb{C}_1^N$ by:

$$\text{error}(z,\hat{z}) = 1 - \frac{|z^*\hat{z}|}{N}. \tag{IV.2}$$

We note that $\text{error}(z,\hat{z}) = 0$ when $\hat{z} = ze^{\iota\phi}$ for any $\phi \in [0,2\pi)$. The error is invariant to a global phase, since $\text{error}(z,e^{\iota\phi}\hat{z}) = \text{error}(z,\hat{z})$ for any $\phi \in [0,2\pi)$.

The network was trained using a data set of dimension $N = 20$ and $M = 20000$ samples generated according to the U(1) Gaussian model. We used the Adam optimizer with batch



(a) $\lambda = 1.2$ (b) $\lambda = 1.5$ (c) $\lambda = 2$

FIGURE 2. Alignment error (IV.1) as a function of depth for the $\mathbb{Z}/2$ synchronization problem with different λ values. The unrolled algorithm is compared against the power method (PM), projected power method (PPM), and the AMP algorithm described in Section II-A. The unrolled synchronization network outperforms the alternative algorithms, and the gap increases as the SNR decreases.

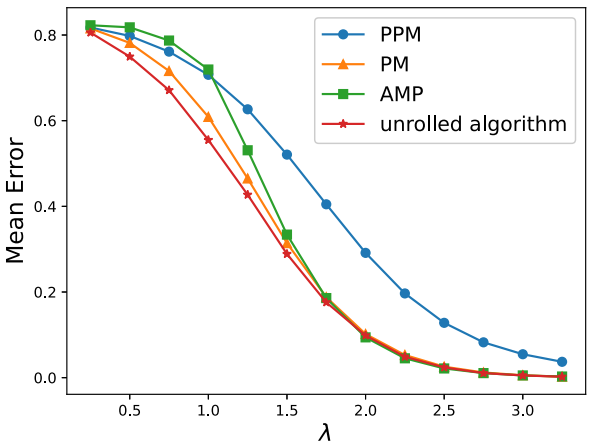


FIGURE 3. Alignment error (IV.1) as a function of the SNR for $\mathbb{Z}/2$ synchronization. The depth of the unrolled algorithm is fixed to 9, while the alternative algorithms ran for 100 iterations. Nevertheless, the unrolled algorithm clearly outperforms the iterative methods.

size of 128, 300 epochs, and a learning rate of 10^{-4}. The results are presented in Fig. 4 for $\lambda = 1.2$, $\lambda = 1.5$ and $\lambda = 2$. As in the $\mathbb{Z}/2$ synchronization, the unrolled synchronization network outperforms the alternative algorithms, especially as the SNR decreases.

C. SO(3) SYNCHRONIZATION

For a ground truth matrix $R \in \mathbb{R}^{3N\times 3}$ (composed of N, 3×3 rotation matrices) and a prediction $\hat{R} \in \mathbb{R}^{3N\times 3}$, the error is defined as:

$$\text{error}(R,\hat{R}) = 1 - \frac{3}{N}\left\|R^T\hat{R}\right\|_{\text{F}}^2. \tag{IV.3}$$

This error metric is invariant to a right multiplication by an orthogonal matrix, and is equal to zero if $\hat{R}$ is equal to R (up to a global rotation).

The network was trained using a dataset of dimension $N = 20$ and $M = 10000$ samples generated according to the model in (II.12). We used the Adam optimizer with batch size of 128, 300 epochs, and a learning rate of 10^{-2}. The spectral method

TABLE 1 Run-Time for SO(3) Synchronization With a Batch of 10000 Samples, $\lambda = 1.5$ $N = 20$ and $L = 9$ Layers, Compared Against PPM With 100 Iterations and the Spectral Method

Algorithm	Alignment Error	Total run-time [s]	Iteration run-time [s]
Spectral method	0.44	18.96	-
Projected power method	0.64	37.52	**0.375**
Unrolled synchronization	**0.22**	**3.53**	0.392

computed the first three eigenvectors of the measurement matrix using SVD factorization as described in II-C1. Therefore, its error is not a function of the number of iterations. The results are presented in Fig. 5, demonstrating a substantial gap between the unrolled algorithm and the competitors.

Fig. 6 shows the error as a function of the SNR, when the depth of the network was fixed to 9, while the projected power method ran for 100 iterations. Nevertheless, the unrolled algorithm clearly outperforms the other methods.

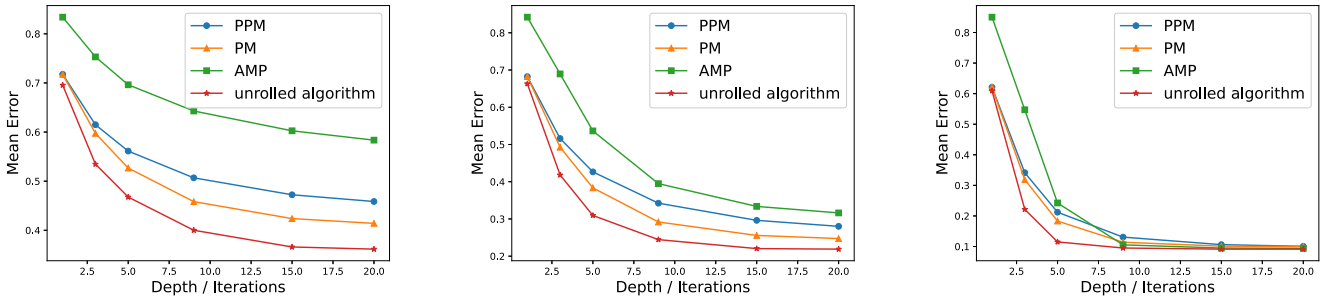
In addition, we measured the inference run-time of a batch of 10000 samples with $\lambda = 1.5, N = 20$ and $L = 9$ layers, and compared it against PPM with 100 iterations and the spectral method. The results are summarized in Table 1. The unrolled algorithm outperforms both methods in terms of alignment error and total run-time due to its low number of layers, with only a slight increase in run-time per layer.

D. MULTI-REFERENCE ALIGNMENT OVER $\mathbb{Z}/2$

We generated measurements according to (II.15) with a signal length of 21, where each entry was drawn i.i.d. from $\mathcal{N}(0,1)$, and $N = 20$. The relative ratios were estimated according to (II.16). In the first part, we evaluate the alignment error (IV.1) using the network described in Section III-A. In the next part, we evaluate the reconstruction error, defined as:

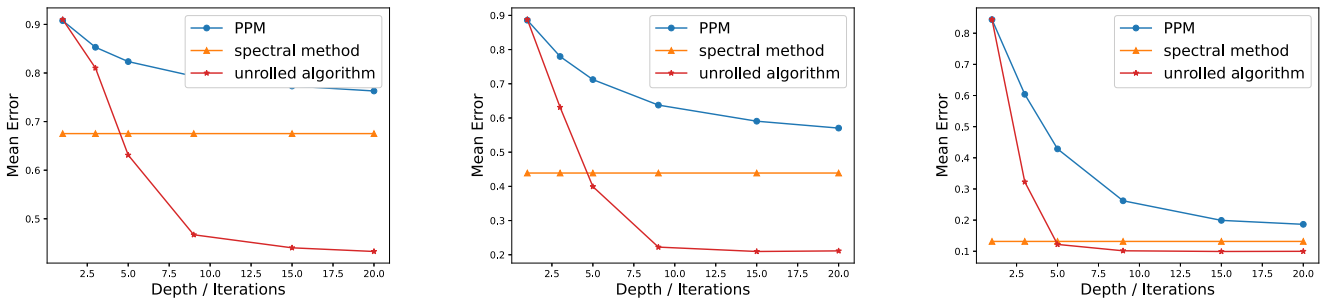
$$\text{error}(x,\hat{x}) = \min_{s\in\{-1,1\}} \|x - s\hat{x}\|^2, \tag{IV.4}$$

where $\hat{x}$ is the estimated signal, computed by aligning the measurements according to the estimated group elements and averaging, as described in (II.17). In this case, we used a modified loss function as described in Section III-D1.



(a) $\lambda = 1.2$ (b) $\lambda = 1.5$ (c) $\lambda = 2$

FIGURE 4. Alignment error (IV.2) as a function of depth for the $SO(3)$ synchronization problem with different λ values. The unrolled algorithm is compared against the power method (PM), projected power method (PPM), and the AMP algorithm described in Section II-B.



(a) $\lambda = 1.2$ (b) $\lambda = 1.5$ (c) $\lambda = 2$

FIGURE 5. Alignment error (IV.3) as a function of depth for the $SO(3)$ synchronization problem with different λ values. The unrolled algorithm is compared against the spectral method and the projected power method (PPM) as described in Section II-C. We note that the spectral method computes the eigenvectors using SVD factorization, and thus the error is not a function of the number of iterations.

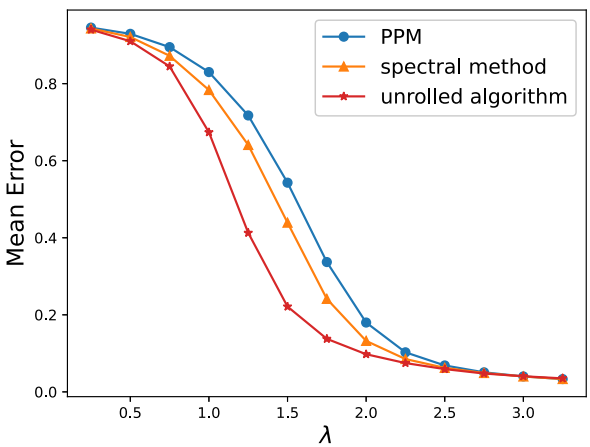


FIGURE 6. Alignment error (IV.3) as a function of the SNR for $SO(3)$ synchronization. The depth of the unrolled algorithm is fixed to 9, while the alternative algorithms ran for 100 iterations. Nevertheless, the unrolled algorithm clearly outperforms the iterative methods.

1) WITH ALIGNMENT LOSS (III.5)

The network was trained using a dataset of $M = 10000$ samples, a batch size of 128, with 300 epochs, and a learning rate of 10^{-4}, using the Adam optimizer. The average alignment error as a function of depth is presented in Fig. 7, for $\lambda = 0.2$ and $\lambda = 0.3$. The experiment shows that the unrolled synchronization network usually achieves better error performance, but the gap is insignificant.

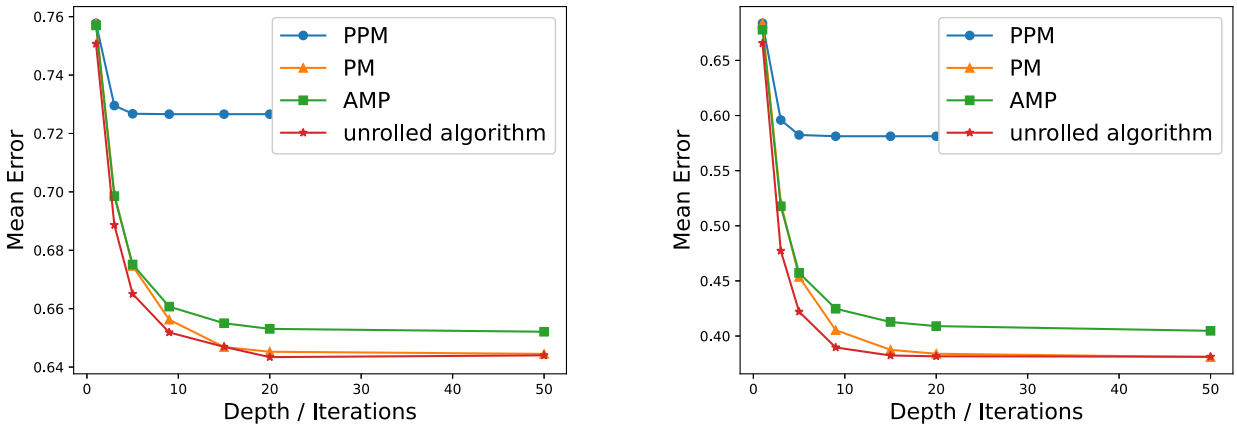
2) WITH RECONSTRUCTION LOSS (III.12)

The network was trained using a dataset of $M = 10000$ samples, a batch size of 128, with 300 epochs, and a learning rate of 10^{-3}, using the Adam optimizer. The average reconstruction error as a function of depth is presented in Fig. 8 for $\lambda = 0.4$ and $\lambda = 0.8$. The experiment shows that the unrolled synchronization network achieves better reconstruction error performance per depth, and outperforms the existing methods for a large number of iterations.

E. MULTI-REFERENCE ALIGNMENT OVER THE GROUP $\mathbb{Z}/L$ OF CIRCULAR SHIFTS

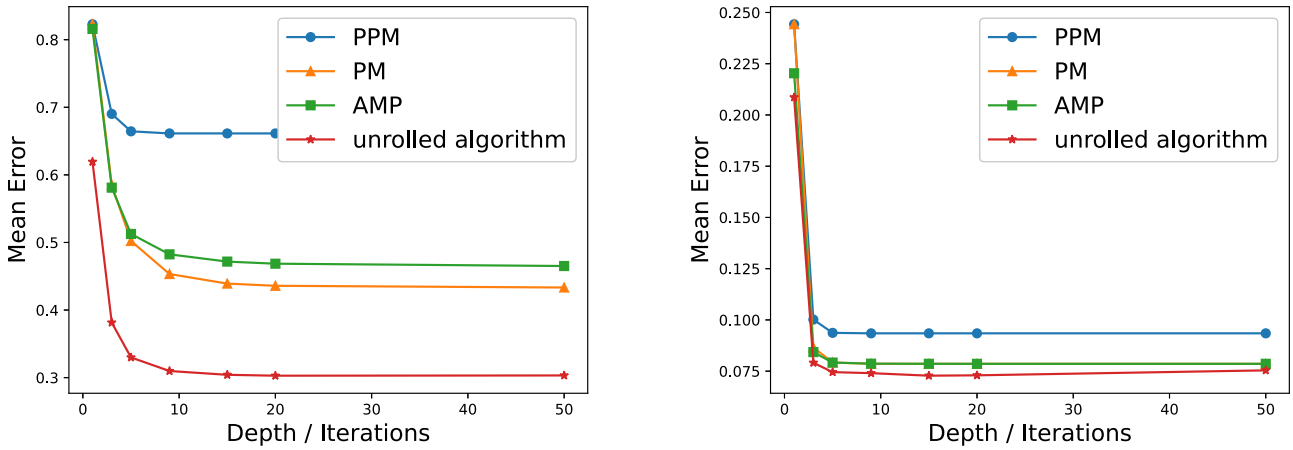
We generated measurements according to (II.19) with a signal of length 21, where each element was drawn i.i.d. from $\mathcal{N}(0, 1)$, and $N = 20$. The relative ratios were estimated according to (II.20) and (II.21). In the first part, we evaluate the alignment error (IV.2) using the network described in Section III-B. In the next part, we evaluate the signal reconstruction error, defined as:

$$\text{error}(\mathcal{X}, \hat{\mathcal{X}}) = \min_{\phi \in \{\frac{2\pi}{L^p}, 2\frac{2\pi}{L^p}, \ldots, 2\pi\}} \|\mathcal{X} - e^{j\bar{k}\phi} \cdot \hat{\mathcal{X}}\|^2, \quad \text{(IV.5)}$$



(a) $\lambda = 0.2$ (b) $\lambda = 0.3$

FIGURE 7. Alignment error (IV.1) as a function of depth for the multi-reference alignment over $\mathbb{Z}/2$ problem with different λ values. The unrolled algorithm is compared against the power method (PM), projected power method (PPM), and the AMP algorithm described in Section II-D1.



(a) $\lambda = 0.4$ (b) $\lambda = 0.8$

FIGURE 8. Reconstruction error (IV.4) as a function of depth for the multi-reference alignment over $\mathbb{Z}/2$ problem with different λ values. The unrolled algorithm, trained using the reconstruction loss function described in Section III-D1, is compared against the power method (PM), projected power method (PPM), and the AMP algorithm described in Section II-D1.

where $\bar{k}$ is the frequency vector at each entry, $\cdot$ is an entry-wise product, and $\hat{\mathcal{X}}$ is the estimated signal in Fourier space, computed by aligning the measurements according to the estimated group elements and averaging, as described in (III.13) and (III.14). In this case, we used a modified loss function as described in Section III-D2. We set $P = 10$.

1) WITH ALIGNMENT LOSS (III.8)

The network was trained using a dataset of $M = 10000$ samples, a batch size of 128, with 300 epochs, and a learning rate of 10^{-4}, using the Adam optimizer. The average alignment error as a function of depth is presented in Fig. 9 for $\lambda = 0.7$. The experiment shows that the error of the unrolled synchronization network improves with the depth of the network, but does not outperform the existing methods for a large number of iterations.

2) WITH RECONSTRUCTION LOSS (III.15)

The network was trained using a dataset of $M = 10000$ samples, a batch size of 128, with 300 epochs, and a learning rate of 10^{-1}, using the Adam optimizer. The average alignment error as a function of depth is presented in Fig. 10 for $\lambda = 1$. The experiment shows that the unrolled synchronization network clearly outperforms the existing methods for a large number of iterations.

F. COMPARING THE UNROLLING ALGORITHM TO PURELY DATA DRIVEN NEURAL NETWORK IN MULTI-REFERENCE ALIGNMENT OVER $\mathbb{Z}/2$ PROBLEM

In this experiment, we compared the suggested unrolled algorithm to an end-to-end trained multi-layer perceptron (MLP), without the inductive bias that we derived from the similarity to the AMP solution. Specifically, we tested the MRA problem over $\mathbb{Z}/2$ using the alignment loss, with $N = 20$, data

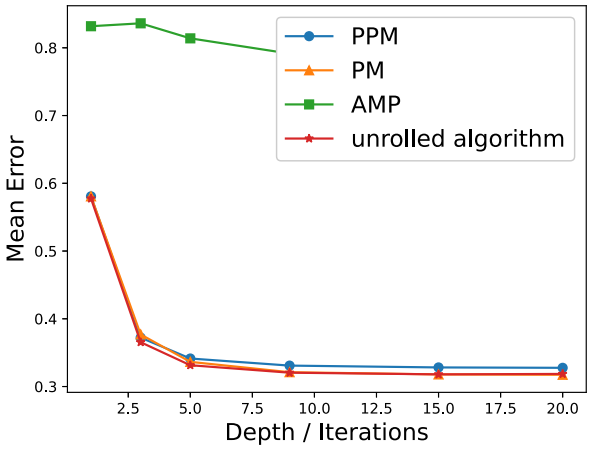


FIGURE 9. Alignment error (IV.1) as a function of depth for the multi-reference alignment over $\mathbb{Z}/L$ with $\lambda = 0.7$. The unrolled algorithm is compared against the power method (PM), projected power method (PPM), and the AMP algorithm described in Section II-D2.

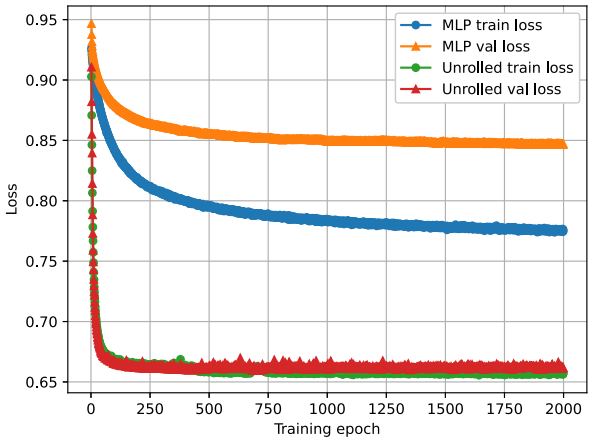


FIGURE 11. Train and validation losses for MRA over $\mathbb{Z}/2$, for the suggested unrolled architecture, compared against a standard MLP architecture.

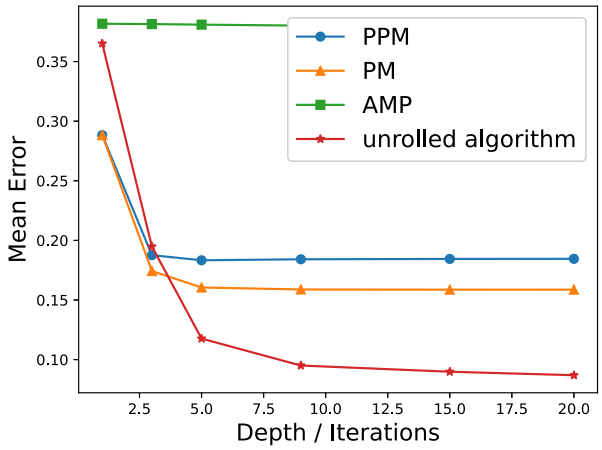


FIGURE 10. Reconstruction error (IV.5) as a function of depth for the multi-reference alignment over $\mathbb{Z}/L$ problem with $\lambda = 1$. The unrolled algorithm, trained using the reconstruction loss function described in Section III-D2, is compared against the power method (PM), projected power method (PPM), and the AMP algorithm described in Section II-D2.

set of 10^4 samples, depth of 5 layers, 2000 epochs, $\lambda = 0.2$ and $L = 21$. Both architectures use a mini-batch size of 128 samples with a learning rate of 10^{-4}. The unrolled network is compared against an MLP network, which receives as an input the relative measurements matrix, reshapes it into a vector of length N^2, projects it into an embedding of length 20 using a basic layer comprises of $Dense(20) \to BN() \to ReLU$. The basic layer is then repeated 5 times, and then projected back with another basic layer to the output dimension of size N. Finally, the output is passed through a $tanh$ activation. Fig. 11 demonstrates how the unrolled algorithm converges faster to a lower loss, and presents smaller difference between the train and validation losses, which suggests that it is not overfitting. We emphasize that the MLP is only one possible architecture, so the conclusion of this experiment should be taken with cautious.

V. DISCUSSION

In this article, we have presented a new computational framework for the group synchronization problem, based on unrolling existing synchronization algorithms, and optimizing them using training data. We introduced unrolling strategies to a wide variety of group synchronization problems, trained using a differentiable invariant synchronization loss function that measures the alignment of the ground truth and the predicted group elements. We have shown that the designed algorithms outperform existing methods for group synchronization. For SO(3) synchronization, we suggested a differentiable feed-forward approximation for the projection operation, which enables training the unrolled algorithm. For the MRA problem, the proposed algorithm incorporates signal prior into the unrolling synchronization algorithm, since the training data consists of relative rotations estimated from noisy measurements drawn from a known distribution.

In the $\mathbb{Z}/2$ synchronization problem, we have demonstrated how the suggested method achieves lower alignment error in the low and moderate SNR regimes, with fewer iterations. In the high SNR regime, the performance of all algorithms is comparable, but the unrolled algorithm still achieves a smaller error per iteration. We then conclude that the proposed method is beneficial for lower SNR regimes, and when running time is a major concern. While existing methods have asymptotic error guarantees, our experiments demonstrate that for a fixed and small number of samples the unrolled synchronization is favorable. We believe that the improved performance stems from our general strategy to optimize existing algorithms (such as AMP) using training data. Moreover, the unrolling synchronization can be readily applied to other noise models, beyond the Gaussian model.

An interesting question is to examine whether a similar technique can be designed for the non-unique games problem: a general optimization framework over groups that can be interpreted as a generalization of the group synchronization problem [42].

Our ultimate goal is to apply our unrolled SO(3) algorithm to cryo-EM experimental data sets. To train the network, in addition to simulated data as in this article, we intend to use experimental data of previously resolved structures available in public repositories [49], and structures resolved using computational tools such as AlphaFold [50].

Another possible future research thread is replacing the unrolling strategy with deep equilibrium (DEQ) to enable a forward model corresponding to an infinite number of layers [51]. Although DEQ models were developed for sequence modeling, they may fit the group synchronization problem: the input sequence is analog to the relative measurement matrix H that is shared among the layers, and the hidden sequence is analog to the estimated group elements.

REFERENCES

[1] A. Singer, "Angular synchronization by eigenvectors and semidefinite programming," *Appl. Comput. Harmon. Anal.*, vol. 30, no. 1, pp. 20–36, 2011.

[2] N. Boumal, "Nonconvex phase synchronization," *SIAM J. Optim.*, vol. 26, no. 4, pp. 2355–2377, 2016.

[3] A. S. Bandeira, N. Boumal, and A. Singer, "Tightness of the maximum likelihood semidefinite relaxation for angular synchronization," *Math. Program.*, vol. 163, no. 1, pp. 145–167, 2017.

[4] Y. Zhong and N. Boumal, "Near-optimal bounds for phase synchronization," *SIAM J. Optim.*, vol. 28, no. 2, pp. 989–1016, 2018.

[5] A. Perry, A. S. Wein, A. S. Bandeira, and A. Moitra, "Message-passing algorithms for synchronization problems over compact groups," *Commun. Pure Appl. Math.*, vol. 71, no. 11, pp. 2275–2322, 2018.

[6] O. Özyeşil, V. Voroninski, R. Basri, and A. Singer, "A survey of structure from motion," *Acta Numerica*, vol. 26, pp. 305–364, 2017.

[7] M. Cucuringu, Y. Lipman, and A. Singer, "Sensor network localization by eigenvector synchronization over the Euclidean group," *ACM Trans. Sensor Netw.*, vol. 8, no. 3, pp. 1–42, 2012.

[8] S. Marchesini, Y.-C. Tu, and H.-T. Wu, "Alternating projection, ptychographic imaging and phase synchronization," *Appl. Comput. Harmon. Anal.*, vol. 41, no. 3, pp. 815–851, 2016.

[9] M. A. Iwen, A. Viswanathan, and Y. Wang, "Fast phase retrieval from local correlation measurements," *SIAM J. Imag. Sci.*, vol. 9, no. 4, pp. 1655–1688, 2016.

[10] T. Bendory, Y. C. Eldar, and N. Boumal, "Non-convex phase retrieval from STFT measurements," *IEEE Trans. Inf. Theory*, vol. 64, no. 1, pp. 467–484, Jan. 2017.

[11] M. Cucuringu, "Sync-rank: Robust ranking, constrained ranking and rank aggregation via eigenvector and SDP synchronization," *IEEE Trans. Netw. Sci. Eng.*, vol. 3, no. 1, pp. 58–79, Jan.–Mar. 2016.

[12] E. Abbe, "Community detection and stochastic block models: Recent developments," *J. Mach. Learn. Res.*, vol. 18, no. 1, pp. 6446–6531, 2017.

[13] D. M. Rosen, L. Carlone, A. S. Bandeira, and J. J. Leonard, "SE-Sync: A certifiably correct algorithm for synchronization over the special Euclidean group," *Int. J. Robot. Res.*, vol. 38, no. 2/3, pp. 95–125, 2019.

[14] O. Ozyesil, N. Sharon, and A. Singer, "Synchronization over Cartan motion groups via contraction," *SIAM J. Appl. Algebra Geometry*, vol. 2, no. 2, pp. 207–241, 2018.

[15] J. Briales and J. Gonzalez-Jimenez, "Cartan-sync: Fast and global SE (d)-synchronization," *IEEE Robot. Automat. Lett.*, vol. 2, no. 4, pp. 2127–2134, Oct. 2017.

[16] T. Bendory, I. Hadi, and N. Sharon, "Compactification of the rigid motions group in image processing," *SIAM J. Imag. Sci.*, vol. 15, no. 3, pp. 1041–1078, 2022.

[17] T. Bendory, D. Edidin, W. Leeb, and N. Sharon, "Dihedral multi-reference alignment," *IEEE Trans. Inf. Theory*, vol. 68, no. 5, pp. 3489–3499, May 2022.

[18] S. Ling, "Near-optimal performance bounds for orthogonal and permutation group synchronization via spectral methods," *Appl. Comput. Harmon. Anal.*, vol. 60, pp. 20–52, 2022.

[19] T. Bendory, A. Bartesaghi, and A. Singer, "Single-particle cryo-electron microscopy: Mathematical theory, computational challenges, and opportunities," *IEEE Signal Process. Mag.*, vol. 37, no. 2, pp. 58–76, Mar. 2020.

[20] I. Greenberg and Y. Shkolnisky, "Common lines modeling for reference free ab-initio reconstruction in cryo-EM," *J. Struct. Biol.*, vol. 200, no. 2, pp. 106–117, 2017.

[21] B. K. Vainshtein and A. B. Goncharov, "Determination of the spatial orientation of arbitrarily arranged identical particles of unknown structure from their projections," *Sov. Phys. Doklady*, vol. 31, 1986, Art. no. 278.

[22] M. V. Heel, "Angular reconstitution: A posteriori assignment of projection directions for 3D reconstruction," *Ultramicroscopy*, vol. 21, no. 2, pp. 111–123, 1987.

[23] A. Singer and Y. Shkolnisky, "Three-dimensional structure determination from common lines in cryo-EM by eigenvectors and semidefinite programming," *SIAM J. Imag. Sci.*, vol. 4, no. 2, pp. 543–572, 2011.

[24] Y. Shkolnisky and A. Singer, "Viewing direction estimation in cryo-EM using synchronization," *SIAM J. Imag. Sci.*, vol. 5, no. 3, pp. 1088–1110, 2012.

[25] V. Monga, Y. Li, and Y. C. Eldar, "Algorithm unrolling: Interpretable, efficient deep learning for signal and image processing," *IEEE Signal Process. Mag.*, vol. 38, no. 2, pp. 18–44, Mar. 2021.

[26] K. Gregor and Y. LeCun, "Learning fast approximations of sparse coding," in *Proc. 27th Int. Conf. Int. Conf. Mach. Learn.*, 2010, pp. 399–406.

[27] Y. Yang, J. Sun, H. Li, and Z. Xu, "ADMM-CSNet: A deep learning approach for image compressive sensing," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 3, pp. 521–538, Mar. 2020.

[28] Y. Li, M. Tofighi, V. Monga, and Y. C. Eldar, "An algorithm unrolling approach to deep image deblurring," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2019, pp. 7675–7679.

[29] Y. Chen and T. Pock, "Trainable nonlinear reaction diffusion: A flexible framework for fast and effective image restoration," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1256–1272, Jun. 2017.

[30] D. Wu, K. Kim, and Q. Li, "Computationally efficient deep neural network for computed tomography image reconstruction," *Med. Phys.*, vol. 46, no. 11, pp. 4763–4776, Nov. 2019.

[31] I. Y. Chun, Z. Huang, H. Lim, and J. Fessler, "Momentum-Net: Fast and convergent iterative neural network for inverse problems," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 4, pp. 4915–4931, Apr. 2023.

[32] A. Mehranian and A. J. Reader, "Model-based deep learning pet image reconstruction using forward–backward splitting expectation-maximization," *IEEE Trans. Radiat. Plasma Med. Sci.*, vol. 5, no. 1, pp. 54–64, Jan. 2021.

[33] S. Chen, Y. C. Eldar, and L. Zhao, "Graph unrolling networks: Interpretable neural networks for graph signal denoising," *IEEE Trans. Signal Process.*, vol. 69, pp. 3699–3713, 2021.

[34] Y. B. Sahel, J. P. Bryan, B. Cleary, S. L. Farhi, and Y. C. Eldar, "Deep unrolled recovery in sparse biological imaging: Achieving fast, accurate results," *IEEE Signal Process. Mag.*, vol. 39, no. 2, pp. 45–57, Mar. 2022.

[35] A. S. Bandeira, M. Charikar, A. Singer, and A. Zhu, "Multireference alignment using semidefinite programming," in *Proc. 5th Conf. Innovations Theor. Comput. Sci.*, 2014, pp. 459–470.

[36] T. Bendory, N. Boumal, C. Ma, Z. Zhao, and A. Singer, "Bispectrum inversion with application to multireference alignment," *IEEE Trans. Signal Process.*, vol. 66, no. 4, pp. 1037–1050, Feb. 2018.

[37] A. S. Bandeira, J. Niles-Weed, and P. Rigollet, "Optimal rates of estimation for multi-reference alignment," *Math. Statist. Learn.*, vol. 2, no. 1, pp. 25–75, 2020.

[38] A. Perry, J. Weed, A. S. Bandeira, P. Rigollet, and A. Singer, "The sample complexity of multireference alignment," *SIAM J. Math. Data Sci.*, vol. 1, no. 3, pp. 497–517, 2019.

[39] E. Abbe, T. Bendory, W. Leeb, J. M. Pereira, N. Sharon, and A. Singer, "Multireference alignment is easier with an aperiodic translation distribution," *IEEE Trans. Inf. Theory*, vol. 65, no. 6, pp. 3565–3584, Jun. 2019.

[40] C. Ma, T. Bendory, N. Boumal, F. Sigworth, and A. Singer, "Heterogeneous multireference alignment for images with application to 2D classification in single particle reconstruction," *IEEE Trans. Image Process.*, vol. 29, pp. 1699–1710, 2020.
[41] N. Boumal, T. Bendory, R. R. Lederman, and A. Singer, "Heterogeneous multireference alignment: A single pass approach," in *Proc. IEEE 52nd Annu. Conf. Inf. Sci. Syst.*, 2018, pp. 1–6.
[42] A. S. Bandeira, Y. Chen, R. R. Lederman, and A. Singer, "Non-unique games over compact groups and orientation estimation in cryo-EM," *Inverse Problems*, vol. 36, no. 6, 2020, Art. no. 064002.
[43] S. Afonso et al., "Estimation under group actions: Recovering orbits from invariants," *Appl. Comput. Harmon. Anal.*, vol. 66, pp. 236–319, 2023.
[44] E. Romanov, T. Bendory, and O. Ordentlich, "Multi-reference alignment in high dimensions: Sample complexity and phase transition," *SIAM J. Math. Data Sci.*, vol. 3, no. 2, pp. 494–523, 2021.
[45] A. Abas, T. Bendory, and N. Sharon, "The generalized method of moments for multi-reference alignment," *IEEE Trans. Signal Process.*, vol. 70, pp. 1377–1388, 2022.
[46] N. Janco and T. Bendory, "An accelerated expectation-maximization algorithm for multi-reference alignment," *IEEE Trans. Signal Process.*, vol. 70, pp. 3237–3248, 2022.
[47] Y. Deshpande, E. Abbe, and A. Montanari, "Asymptotic mutual information for the binary stochastic block model," in *Proc. IEEE Int. Symp. Inf. Theory*, 2016, pp. 185–189.
[48] Å. Björck and C. Bowie, "An iterative algorithm for computing the best estimate of an orthogonal matrix," *SIAM J. Numer. Anal.*, vol. 8, no. 2, pp. 358–364, 1971.
[49] A. Iudin, P. K. Korir, J. Salavert-Torres, G. J. Kleywegt, and A. Patwardhan, "EMPIAR: A public archive for raw electron microscopy image data," *Nature Methods*, vol. 13, no. 5, pp. 387–388, 2016.
[50] J. Jumper et al., "Highly accurate protein structure prediction with AlphaFold," *Nature*, vol. 596, no. 7873, pp. 583–589, 2021.
[51] S. Bai, J. Z. Kolter, and V. Koltun, "Deep equilibrium models," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, pp. 690–701.