Quantum Permutation Synchronization

Tolga Birdal1,* Vladislav Golyanik2,* Christian Theobalt2 Leonidas Guibas1

1Stanford University 2Max Planck Institute for Informatics, SIC

Abstract

We present QuantumSync, the first quantum algorithm for solving a synchronization problem in the context of computer vision. In particular, we focus on permutation synchronization which involves solving a non-convex optimization problem in discrete variables. We start by formulating synchronization into a quadratic unconstrained binary optimization problem (QUBO). While such formulation respects the binary nature of the problem, ensuring that the result is a set of permutations requires extra care. Hence, we: (i) show how to insert permutation constraints into a QUBO problem and (ii) solve the constrained QUBO problem on the current generation of the adiabatic quantum computers D-Wave. Thanks to the quantum annealing, we guarantee global optimality with high probability while sampling the energy landscape to yield confidence estimates. Our proof-of-concepts realization on the adiabatic D-Wave computer demonstrates that quantum machines offer a promising way to solve the prevalent yet difficult synchronization problems.

1. Introduction

Computer vision literature accommodates a myriad of efficient and effective methods for processing rich information from widely available 2D and 3D cameras. Many algorithms at our disposal excel at processing single frames or sequence of images such as videos [82, 34, 13]. Oftentimes, a scene can be observed by multiple cameras, from different, usually unknown viewpoints. To this date it remains an open question how to consistently combine the multi-view cues acquired without respecting any particular order [44].

Synchronization [91, 42, 92] is one of the proposed solutions to the aforementioned problem. On an abstract note, it involves distributing the discrepancies over the graph connecting multiple viewpoints such that the estimates are consistent across all considered nodes. To this end, synchronization simultaneously averages the pairwise local information into a global one [48, 54, 4, 12]. This procedure is a fundamental piece of most state-of-the-art multi-view reconstruction and multi-shape analysis pipelines [85, 23, 25] because it heavy-lifts the global constraint satisfaction while respecting the geometry of the parameters. In fact, most of the multiview-consistent inference problems can be expressed as some form of a synchronization [107, 15].

In this paper, our focus is permutation synchronization, where the edges of the graph are labeled by permutation matrices denoting the correspondences either between two 2D images or two 3D shapes. Specifically, we seek to find an absolute ordering for each point of each frame which sorts all the corresponding points into the same bin. Unfortunately, this problem by definition involves a combinatorial non-convex optimization, for which attaining the global minimum is intractable under standard formulations targeting classical von Neumann computers. This difficulty have encouraged scholars to seek continuous relaxations for which either a good local optimum [14, 16] or a closed form solution [6, 57, 5] could be found. The solution to this approximately equivalent relaxed problem should then be rounded back to the permutation matrices to report a valid result. Ideally, we would like to avoid such approximations and work with the original set of constraints at hand. This is exactly what we propose to do in this work. Because on a classical computer we cannot speak of a global optimality guarantee of our discrete problem, we turn our attention to a new family of processors, i.e., quantum computers.

* both authors contributed equally to this work
A quantum computer is a computing machine which takes advantage of quantum effects such as quantum superposition, entanglement, tunnelling and contextuality to solve problems notoriously difficult (i.e., $NP$-hard) on a classical computer. Numerous quantum algorithms have demonstrated improved computational complexity finally reaching the desired supremacy [7]. Currently, we see the first practical uses of quantum computers thanks to the programmable quantum processing units (QPU) available to the research community such as D-Wave [2] or IBM [1]. As this new generation of computers offers a completely different computing paradigm, directly porting classical problem formulations is far from being trivial. In fact, oftentimes we are required to revise the problem at hand altogether [49].

This paper shows how to formulate the classical permutation synchronization in terms of a quadratic unconstrained binary optimization (QUBO) that is quantum computer friendly. QUBO optimizes for binary variables and not permutations, and, hence, we are required to ensure that permutation constraints are respected during optimization. To this end, we turn all constraints into linear ones and incorporate them into QUBO. Finally, we embed our problem to a real quantum computer and show that it is highly likely to achieve the lowest energy solution, the global optimum (see Fig. 1). Our contributions are as follows:

1. The first, to the best of our knowledge, formulation of the classical synchronization in a form consumable by an adiabatic quantum computer (AQC);
2. We show how to introduce permutations as linear constraints into the QUBO problem;
3. We numerically verify the validity of our formulation in simulated experiments as well as on a real AQC (for the first time, on D-Wave Advantage 1.1);
4. We perform extensive ablation studies giving insights into this new way of computing.

We obtain highly probable global optima up to either of: (i) eight views and three points per view, (ii) seven views with four points, or (iii) five points with three views. We experiment for the first time on an AQC with 5$k$ qubits and perform extensive evaluations and comparisons to classical methods. Our approach can also be classified as the first method for quantum matching of multiple point sets. While we are the first to implement synchronization on quantum hardware, our evaluations and tests are proof of concepts as truly practical quantum computing is still a leap away.

### 2. Related Work

**Synchronization.** The art of consistently recovering absolute quantities from a collection of ratios, synchronization, is now the de-facto choice when it comes to bringing functions on image/shape collections into unison [85, 23, 25]. The problem is now very well studied, enjoying a rich set of algorithms. Primarily, there exists a plethora of works on group structures arising in different applications [48, 47, 16, 5, 4, 56, 54, 4, 100, 28, 96, 98, 6, 9]. Some of the proposed solutions are closed form [6, 5, 72, 4] or minimize robust losses [53, 27]. Others address certifiability [83] and global optimality [21]. Bayesian treatment or uncertainty quantification is also considered [97, 14, 12, 16] as well as low rank matrix factorizations [10]. Recent algorithms tend to incorporate synchronization into deep learning [58, 43]. This work concerns with synchronizing correspondence sets, otherwise known as permutation synchronization (PS) [81]. This sub-field also attracted a descent amount of attention: low-rank formulations [104, 101], convex programming [55], multi-graph matching [86], distributed optimization [55] or Riemannian optimization [16].

All of these approaches try to cope with the intrinsic non-convexity of the synchronization problem one way or another. Unfortunately, solving our problem on a classical computer is notoriously difficult. To the best of our knowledge, we are the firsts to address this problem through the lens of a new paradigm, adiabatic quantum computing.

**Quantum computing.** Since its motivation in the 1980s [71, 39], quantum computing has become an active research area, both from the hardware [30, 61, 36, 102, 73, 111, 65, 68, 66] and algorithmic side [89, 90, 17, 49, 38, 51, 99, 52, 31, 3, 70, 87, 75, 37, 108, 94]. Quantum methods offering speedup compared to the classical counterparts have been demonstrated for domains such as applied number theory [89, 17, 99, 37], linear algebra [51, 52, 67, 95, 94], machine learning [3, 70, 108] and simulation of physical systems [106, 79, 11, 62, 59], among others.

Quantum annealing (QA) [60, 22] and the adiabatic quantum evolution algorithm by Farhi et al. [38] have triggered the development of adiabatic quantum computers (AQC). In the last decade, the technology has matured and became accessible remotely for test and research purposes with the help of D-Wave [32]. A recent benchmarking of D-Wave AQC [33] has shown that for energy landscapes with large and tall barriers, quantum annealing can achieve speed-ups of up to eight orders of magnitude compared to simulated annealing [63] running on a single core. AQC has also been successfully applied to traffic flow optimization while outperforming classical methods [75].

**Quantum computer vision.** Several quantum methods for computer vision problems have been proposed in the literature including algorithms for image recognition [76], classification [20, 78] and facial feature learning by low-rank matrix factorization [80]. Recently, D-Wave has been applied to redundant object removal in object detection [69]. While a QUBO formulation of non-maximum suppression was already known in the literature [84], it has been improved in [69] for D-Wave 2X with solutions outperforming several classical methods. Quantum approach for point
set alignment [45, 46] solves a related problem to ours. It approximates rotation matrices in the affine space by basis elements which can be summed up according to the measured bitstring. However, this formulation cannot be easily extended to the multi-view case and does not support permutation constraints. In contrast, our method solves a multi-way matching problem, and we successfully deploy it on a real AQC Advantage 1.1. Our use of assignment constraints is also new. Concurrently to us, a similar form of permutation matrix constraints was proposed by Benkner et al. [88] for matching two graphs on AQC. Several previous works formulate similar penalties in different ways. Stollenwerk et al. [93] address the flight assignment problem on AQC. Formulation of the graph isomorphism problem can include permutations [110, 41]. In [110], an individual variable for every vertex pair of the same degree in a graph is allocated, which subsequently leads to a QUBO. Permutations can also be converted to a table with binary entries added as a penalty term to the target Hamiltonian [41].

3. Preliminaries and Technical Background

3.1. Synchronization

We model the multi-view configuration as a connected undirected graph \( \mathcal{G} = (\mathcal{V} = \{1, 2, \ldots, n\}, \mathcal{E} \subset [n] \times [n]) \) where \(|\mathcal{E}| = m\) and if \((i, j) \in \mathcal{E}\) then \((j, i) \in \mathcal{E}\). Each vertex \(v_i \in \mathcal{V}\) (e.g., image) is associated a domain \(D_i\) (e.g., ordered points). Each edge \((i, j) \in \mathcal{E}\) is labeled with a function \(f_{i,j} : D_i \mapsto D_j\) (e.g., correspondence). We will refer to the edge-related entities as relative and node- (vertex-) related entities as absolute. Thus, we aptly call \(\{f_{i,j}\}_{i,j}\) as relative maps. In this section, we define and explain the necessary notions following Fig. 2. The proofs of the theorems given in this section can be found in our supplementary material.

Definition 1 (Path, Cycle and Null Cycle). We define a path to be the ordered, unique index sequence \(p = \{(i_1, i_2), (i_2, i_3), \ldots, (i_{n-1}, i_n)\} \in \mathcal{P}\) along \(\mathcal{G}\) connecting \(v_{i_1}\) to \(v_{i_n}\). It is called a cycle, if additionally the path traces back to the starting node. Finally, following [4], we denote a cycle \(c \in \mathcal{C}\) to be a null-cycle of \(\mathcal{G}\) if the composition of functions along \(c\) leads to the identity transformation:

\[
    f_c = f_{1,2} \circ f_{2,3} \cdots \circ f_{(n-1),n} \circ f_{n,1} = f_\emptyset, \tag{1}
\]

where \(f_\emptyset(x) = x, \forall x \in D_1\) is the identity map and \(f_c\) denotes the composite function. Intuitively, \(c\) is a non-empty path in which the only repeated vertices are the first and last vertices. Note that the direction of the action of \(\circ\) matters (e.g., \(f_{1,2} \circ f_{2,3} \neq f_{2,3} \circ f_{1,2}\)) and is up to the convention.

Definition 2 (k-cycle). We refer to a function mapping a vertex to itself as the 1-cycle: \(f_{i,i} = f_\emptyset\). Similarly, a 2-cycle would be \(f_c = f_{i,j} \circ f_{j,i}\) and so on [77, 57].

![Figure 2. (left) Notation and illustration of the general setting we consider. (right) Cycle consistency vs path invariance.](Image)

Figure 2. (left) Notation and illustration of the general setting we consider. (right) Cycle consistency vs path invariance.

Definition 3 (Cycle Consistency). We call the graph \(\mathcal{G}\) to be cycle-consistent on \(\mathcal{C}\) if \(f_c = f_\emptyset \ \forall c \in \mathcal{C}\), where \(\mathcal{C}\) is the set of all cycles [50].

The notion of cycle consistency for directed graphs is known as path invariance [107] and differs from cycle consistency as shown in Fig. 2. In this paper, we further assume the maps belong to the general linear group and are isomorphisms, i.e., \(f_{i,j} = f_{j,i}^{-1}\).

Remark 1. Depending on the graph topology, the number of cycles may be exponential in the number of vertices. Hence, naively ensuring the consistency of large graphs according to Dfn. 3 quickly becomes intractable. Algorithms such as Guibas et al. [50] aim to satisfy the consistency of a subset of cycles \(\mathcal{C} \subset \mathcal{C}\) (bases), where enforcing consistency along these cycles induces consistency along all cycles of the input graph \(\mathcal{G}\). However, as efficient selection of these cycle-consistency bases [105] is a problem under investigation, we instead use the available group structure:

Theorem 1 (Cycle Consistency by Construction). A consistent vertex labeling \(\{f_i : \mathcal{V} \mapsto D_i\}_i\) where \((D, \circ)\) forms a group with operation \(\circ\), can be constructed by satisfying the following constraint for all \(i\) and \(j\):

\[
    f_{i,j} = f_i \circ f_j^{-1}. \tag{2}
\]

Hence, Eq (2) is called the cycle consistency constraint.

Definition 4 (Synchronization). Synchronization is the procedure of finding a consistent labeling of \(\mathcal{G}\) given a collection of ratios \(\{f_{i,j}\}_{i,j}\) ensuring the cycle consistency of \(\mathcal{G}\):

\[
    \arg\min_{\{f_{i,j}\}_i} \sum_{(i,j) \in \mathcal{E}} d(f_{i,j}, f_{i,j}). \tag{3}
\]

\(\hat{f}_{i,j} = f_i \circ f_j^{-1}\) are the estimated ratios and \(d(\cdot)\) is a group-specific distance metric.

Remark 2. A closer look to the problem reveals that it is non-convex due to the composition, but convex when the vertices \(f_j\) are fixed during optimization of \(f_i\). In fact, if
f_i is considered to be fixed, this problem resembles an averaging under the metric d(.). As for different i we have different averages to compute, synchronization is often referred as multiple averaging [40, 54, 26, 53].

**Theorem 2** (Gauge Freedom). The problem in Dfn. 4 is subject to a freedom in the choice of the reference or the gauge [8, 26]. In other words, the solution set to Eq (3) can be transformed arbitrarily by a common f_j i.e. f_i \leftarrow f_i \circ f_g while still satisfying the consistency constraint.

In practice, a gauge is fixed by setting one of the vertex labels to identity: f_1 = f_0.

**Definition 5** (Permutation Matrix). A permutation matrix is defined as a sparse, square binary matrix, where each column or row contains only a single non-zero entry:

\[ P_n = \{ P \in \{0, 1\}^{n \times n} : P1_n = 1_n, \quad 1_n^T P = 1_n^T \} \]  

(4)

where 1_n denotes a n-dimensional ones vector. Every P \in \mathcal{P}_n is a total permutation matrix and P_{ij} = 1 implies that point i is mapped to element j. Note, P_i^T = P_i^{-1}. We also denote the product manifold of \( m \) permutations as \( \mathcal{P}_m \).

**Definition 6** (Relative Permutation). We define a permutation matrix to be a relative map if it is the ratio (or difference) of two group elements \((i \rightarrow j) : P_{ij} = P_i P_j \). (5)

**Definition 7** (Permutation Synchronization Problem). For a redundant set of measures of ratios \{P_{ij}\}, the permutation synchronization [81] seeks to recover \{P_i\} for \( i = 1, \ldots, N \) such that Eq (2) is satisfied: \( P_{ij} = P_i P_j^{-1} \).

If the input data is noise-corrupted, this consistency will not hold and to recover the absolute permutations \{P_i\}, some form of a consistency error is minimized.

### 3.2. Adiabatic Quantum Computation

Contemporary AQC can solve QUBO over a set of pseudo-Boolean functions of the following form:

\[ \arg \min_{x \in B^n} x^T Q x + s^T x, \]

(5)

where \( B^n \) denotes the set of binary vectors of length n, \( Q \in \mathbb{R}^{n \times n} \) is a real symmetric matrix and \( s \) is a real n-dimensional vector. (5) is a frequent problem that is known to be \( \mathcal{NP} \)-hard on a classical computer. AQA operates with qubits obeying the laws of quantum mechanics. In contrast to a classical bit, a qubit \( |\phi\rangle \) can continuously transition between the states \( |0\rangle \) and \( |1\rangle \) (the equivalents of classical states 0 and 1) fulfilling the equation \( |\phi\rangle = \alpha |0\rangle + \beta |1\rangle \), with probability amplitudes satisfying \( |\alpha|^2 + |\beta|^2 = 1 \).

AQA algorithms for problems in a non-QUBO form first have to encode the problem as QUBO (5), which defines the former in terms of logical qubits and weights (s and Q) between them. The logical problem is then minor-embedded to the AQA’s physical qubits graph, with methods such as [24]. AQA interprets s and Q as qubit biases and couplings, respectively, and converts them to local magnetic fields on a QPU imposed during annealings, i.e., free evolutions of the quantum-mechanical computing system. The search of the optimal x is performed by optimizing over the hidden states of n qubits and consequently unembedding and measuring them as classical bitstrings. The latter are then passed to the solution interpretation step, which decodes them in the context of the original problem.

Before annealing, all n qubits are initialized in ground states of an initial Hamiltonian \( \mathcal{H}_I \) that is easy to achieve as a superposition with equal probabilities of measuring \( |0\rangle \) or \( |1\rangle \) for every qubit. A Hamiltonian is an operator defining the energy spectrum of the system and— interpreted for computational problems—the space of all possible solutions. AQA performs a series of annealings, during which \( \mathcal{H}_I \) continuously alters towards the problem Hamiltonian \( \mathcal{H}_P \) under the influence of the magnetic fields. This instantaneous Hamiltonian \( \mathcal{H} \) can be expressed as:

\[ \mathcal{H}(\tau) = [1 - \tau] \mathcal{H}_I + \tau \mathcal{H}_P, \]

(6)

with the local time variable \( \tau \in [0; 1] \) transitioning from 0 to 1 during the annealing time t (e.g., 20 \mu s). According to the adiabatic theorem of quantum mechanics [19], the system will likely remain in the ground state of \( \mathcal{H}_P \) by the end of the anneal, assuming a sufficiently long t and despite a highly non-convex energy landscape of the problem. This is due to the quantum effects of superposition, entanglement and tunneling. Superposition enables the optimization to be performed on all possible qubit states simultaneously; it allows to operate on a \( 2^n \)-dimensional state space spanned by n qubits. During quantum computation, entangled states are created, i.e., the states of multiple qubits which cannot be described independently from each other. Tunnelling enables transition through the barriers (in geometric interpretation). For a more comprehensive overview of the AQC foundations, see [60, 38, 2] as well as Secs. 2 and 3 of [45].

### 4. Quantum Synchronization

Suppose a multi-view configuration where we are given n points in each of the m views. Points in view i relate to the ones in view j via a permutation \( P_{ij} \) (see Sec. 3.1). \( P_{ij} \) is usually obtained independently for each pair in the edge set \( \mathcal{E} \) and hence is noisy. Following [16], we see the permutation synchronization as a probabilistic inference problem, where we will be interested in the following quantities:

1. Maximum a-posteriori (MAP):

\[ X^* = \arg \max_{X \in \mathcal{P}_m} \log p(X|P) \]

(7)

\[ 1 \text{in the sense that } \alpha, \beta \text{ cannot be revealed (measurement postulate) and non-destructive copying of } |\phi\rangle \text{ is not possible (no-cloning theorem [103])} \]
where \( \log p(\mathbf{X} | \mathbf{P}) = + \beta \sum_{(i, j) \in E} \| \mathbf{P}_{ij} - \mathbf{X}_i \mathbf{X}_j^\top \|_F \), and \(+\) denotes equality up to an additive constant.

2. The full posterior distribution: \( p(\mathbf{X} | \mathbf{P}) \propto p(\mathbf{P}, \mathbf{X}) \).

Here, \( \mathbf{X}^* \) denotes the entirety of the sought permutations, and, similarly, \( \mathbf{P} \) is the collection of all specified pairwise permutations. The MAP estimate is often easier to obtain and useful in practice. On the other hand, samples from the full posterior can provide important additional information, such as uncertainty. Not surprisingly, the latter is a much harder task, especially considering the discrete nature of our problem. Each AQC algorithm includes problem encoding, minor embedding, sampling and solution interpretation, as described in Sec. 3.2. In what follows, we will describe the problem encoding and the preparation of \( \mathbf{Q} \) in Eq (5).

### 4.1. Permutation Synchronization as QUBO

We first re-write the synchronization loss in Eq (7) as a quadratic assignment problem (QAP) that is more friendly for adiabatic optimization, and later insert the permutations as linear constraints into the formulation.

**Proposition 1.** Permutation synchronization under the Frobenius norm can be written in terms of a QUBO:

\[
\arg\min_{\{\mathbf{X}_i \in \mathbb{P}_n\}} \sum_{(i, j) \in E} \| \mathbf{P}_{ij} - \mathbf{X}_i \mathbf{X}_j^\top \|_F^2 = \arg\min_{\{\mathbf{X}_i \in \mathbb{P}_n\}} \mathbf{x}^\top \mathbf{Q}' \mathbf{x}.
\]

Here, \( \mathbf{x} = [\cdots \mathbf{x}_i^\top \cdots]^\top \) and \( \mathbf{x}_i = \text{vec}(\mathbf{X}_i) \) where \( \text{vec}(\cdot) \) acts as a vectorizer. \( \mathbf{Q}' \) is then a matrix of the form:

\[
\mathbf{Q}' = -\left[ \begin{array}{cccc}
\mathbf{I} \otimes \mathbf{P}_{11} & \mathbf{I} \otimes \mathbf{P}_{12} & \cdots & \mathbf{I} \otimes \mathbf{P}_{1m} \\
\mathbf{I} \otimes \mathbf{P}_{21} & \mathbf{I} \otimes \mathbf{P}_{22} & \cdots & \mathbf{I} \otimes \mathbf{P}_{2m} \\
\vdots & \vdots & \ddots & \vdots \\
\mathbf{I} \otimes \mathbf{P}_{m1} & \mathbf{I} \otimes \mathbf{P}_{m2} & \cdots & \mathbf{I} \otimes \mathbf{P}_{mm}
\end{array} \right].
\]

**Proof.** The steps are intuitive to follow and an expanded proof is included in the supplementary material:

\[
\mathbf{X}^* = \arg\min_{\mathbf{X} \in \mathbb{P}_n^{mn}} \sum_{(i, j) \in E} \| \mathbf{P}_{ij} - \mathbf{X}_i \mathbf{X}_j^\top \|_F^2
\]

\[
= \arg\min_{\mathbf{X} \in \mathbb{P}_n^{mn}} 2N^2n - 2 \sum_{(i, j) \in E} \text{tr}(\mathbf{X}_j \mathbf{X}_i^\top \mathbf{P}_{ij})
\]

\[
= \arg\min_{\mathbf{X} \in \mathbb{P}_n^{mn}} - \sum_{(i, j) \in E} \text{vec}(\mathbf{X}_i)\top (\mathbf{I} \otimes \mathbf{P}_{ij})\text{vec}(\mathbf{X}_j)
\]

\[
= \arg\min_{\mathbf{X} \in \mathbb{P}_n^{mn}} \mathbf{x}^\top \mathbf{Q}' \mathbf{x}.
\]

The Hessian of this problem is given by \( \mathbf{Q}' \) itself. Hence, the problem is only convex and solvable by algorithms such as interior-point or trust-region methods when \( \mathbf{Q}' \) is positive definite. The indefinite problems can be solved via active-set methods if the indefinite variables are relaxed to the set of reals. For discrete variables and when \( \mathbf{Q}' \) is not positive definite, this is a variant of \( \mathcal{NP} \)-hard integer quadratic problem. We will instead show how to use the recent quantum computers to obtain the global minimum.

**Formulating the QUBO synchronization.** The problem in Eq (11) has an added difficulty of being a discrete combinatorial optimization problem over the product manifold of permutations. Replacing permutations with different choices of matrices lead to different relaxations, for example: (i) positivity allows for semi-definite programming [57], (ii) orthonormality allows for spectral solutions [72], and (iii) doubly stochastic relaxation can allow for Riemannian optimization [16]. However, all of these methods have to be followed by a projection step onto the discrete Permutohedron, often cast as an assignment problem and solved via the celebrated Hungarian algorithm. QUBO enables us to solve this problem without continuous relaxations in a globally optimal manner by rephrasing Eq (11) in terms of binary variables \( \mathbf{B} \) at our disposal:

\[
\arg\min_{\mathbf{x} \in \mathbb{B}} \mathbf{x}^\top \mathbf{Q}' \mathbf{x}.
\]

**Permutations as linear constraints.** The binary variables \( \mathbf{x} \in \mathbb{B} \) are a superset of the product-permutations. In other words, the solution to Eq (11) need not result in permutations once the matrices corresponding to each node are extracted. We propose to encourage the solution towards a set of permutations by introducing linear constraints \( \mathbf{A} \mathbf{x} = \mathbf{b} \) such that the optimization adheres to the definition of a permutation: rows and columns sum to one as in Eq (4). Given \( \mathbf{x}_i = \text{vec}(\mathbf{X}_i) \), this amounts to having \( \mathbf{b}_i = 1 \) and

\[
\mathbf{A}_i = \begin{bmatrix} \mathbf{I} \otimes \mathbf{1}^\top \\ \mathbf{1}^\top \otimes \mathbf{I} \end{bmatrix}.
\]

Put simply, the matrix \( \mathbf{A}_i \) is assembled as follows: in row \( j \) with \( 1 \leq j \leq n \), the ones are placed in columns \( (j - 1) \cdot n + 1 \) to \( (j) \cdot n \). In a row \( j \) with \( j > n \), ones will be placed at \( (j - n) + p \cdot n \) for \( p \in \{0, ..., n - 1\} \). To enforce the permutation-ness of all the individual \( \mathbf{x}_i \), we make that up \( \mathbf{x} \in \mathbb{R}^{n^2 \times m} \), we construct a \( n^2 \times 2n \) block-diagonal matrix \( \mathbf{A} = \text{diag}(\mathbf{A}_1, \mathbf{A}_2, \ldots, \mathbf{A}_m) \).

**Introducing linear constraints into QUBO.** We now extend our formulation by introducing the equality constraints \( \mathbf{A} \mathbf{x} = \mathbf{b} \) into the optimization.

**Proposition 2.** The constrained minimization:

\[
\arg\min_{\mathbf{x} \in \mathbb{B}} \mathbf{x}^\top \mathbf{Q}' \mathbf{x} \quad \text{s.t.} \quad \mathbf{A} \mathbf{x} = \mathbf{b}
\]

can be turned into an (unconstrained) QUBO

\[
\arg\min_{\mathbf{x} \in \mathbb{B}} \mathbf{x}^\top \mathbf{Q} \mathbf{x} + \mathbf{s}^\top \mathbf{x},
\]

where \( \mathbf{Q} = \mathbf{Q}' + \lambda \mathbf{A}^\top \mathbf{A} \) and \( \mathbf{s} = -2\lambda \mathbf{A}^\top \mathbf{b} \).
Both $Q'$ and $Q$ are sparse matrices. We provide their sparsity patterns and the proof of Prop. 2 in our supplement. We finally map this modified QUBO $(Q, s)$ onto D-Wave.

**Encoding, Sampling and Interpretation.** We fix the gauge by letting the first matrix to be identity: $X_1 = I$. Once QuantumSync terminates, the measured bitstring can be directly interpreted as the solution permutations after reordering into $m - 1$ matrices of dimension $n \times n$. To explain the posterior landscape, we propose to sample from many low-energy states using the same quantum annealing. Further details are presented in the supplement.

## 5. Experiments and Evaluations

### Real dataset.** To showcase that quantum computers offer a promising way to solve the challenging multi-view matching problems, we extract four categories (duck, car, winebottle, motorbike) of Willow Object Classes [29] composed of 40 RGB images each, acquired in the wild (see Fig. 3). The images suffer from significant pose, lighting and environment variation. Hence, ten keypoints are manually annotated on each image. We use the first four of these keypoints and create 35 small problems for each category by creating a fully connected graph composed of all four consecutive frames. We follow [101] and extract local features from a set of $227 \times 227$ patches centered around the annotated landmarks, using Alexnet [64] pretrained on ImageNet [35]. The feature map responses of Conv4 and Conv5 layers are then matched by the Hungarian algorithm [74] to initialize the synchronization. As the data is manually annotated, the ground-truth relative maps are known.

**Synthetic dataset.** For a controlled evaluation of our method, similar to [16], we generate synthetic datasets composed of graphs with $m = |\mathcal{V}|$ nodes and $|\mathcal{E}| = m(m - 1)$ edges. At each node $i$ we generate $n$ points mappable by a permutation $P_{ij}$ to $n$ other points at node $j$. Hence, all the permutation matrices $P_{ij}$ or $P_{ji}$ are total i.e., of size $n \times n$. While the graph is by default fully connected, in certain experiments we randomly drop certain edges to have a completeness $C$ where $0.5 < C < 1$. For instance, $C = 0.75$ means that only 75% of the edges are actively present. We optionally perturb the relative permutations by swapping a percentage of the rows and columns. We call this the swap ratio and denote it as $\sigma$, where $0 \leq \sigma \leq 0.25$.

**Evaluation methodology.** We implement our algorithm on D-Wave Advantage 1.1 and compare it against the state-of-the-art methods MatchEIG [72], MatchALS [109], MatchLift [57], MatchBirkhoff [16] as well as to the exhaustive solution obtained by enumerating all possible permutations. In all of our evaluations we report the number of bits correctly detected and call this metric accuracy. For synthetic evaluations involving noise, we generate seven random problems with the same $\sigma$ and average the results.

### 5.1. Evaluations on D-Wave Advantage

In this section, we describe our experiments on D-Wave Advantage system 1.1, which is an AQC with 5436 qubits arranged on a graph of cells with eight qubits each. Every qubit operates under $\approx 15.8\,\text{mK}$ temperature and is connected to 15 other qubits from the same or other cells which enables compact minor embeddings, i.e., mappings of target QUBOs to the processor topology with shorter chains of physical qubits compared to the previous 2000Q [18]. To map logical qubits which are connected to more than 15 other logical qubits, chaining of physical qubits is necessary, i.e., entanglement between qubit states. These chains are maintained by auxiliary magnetic fields and can break during annealings. The mechanism of resolving broken chains is majority voting. Minor embeddings are performed automatically via [24], and each annealing takes $20\mu$s – apart from AQC problem transmission overheads which can sum up to 0.1 sec, minor embedding time (which, however, can be pre-computed for multiple problem sizes) and AQC waiting time. We access D-Wave machines remotely through Leap2 [32]. The total AQC runtime spent in the experiments amounts to $\approx 15.5$ min. ($> 5 \times 10^5$ annealings in total). We show exemplary embeddings in our supplement.

**Evaluation on the real dataset.** We begin by putting D-Wave to test on synchronizing real data. Tab. 1 shows the accuracy of the best (lowest energy) solution we attain for different categories, as well as averaged over all classes. It is seen that while quantum solution is not the top-performer, it is certainly on par with the well engineered approaches of

<table>
<thead>
<tr>
<th>Car</th>
<th>Duck</th>
<th>Motorbike</th>
<th>Winebottle</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.91 ± 0.015</td>
<td>0.86 ± 0.017</td>
<td>0.87 ± 0.018</td>
<td>0.85 ± 0.017</td>
</tr>
<tr>
<td>0.88 ± 0.019</td>
<td>0.87 ± 0.023</td>
<td>0.86 ± 0.019</td>
<td>0.85 ± 0.019</td>
</tr>
<tr>
<td>0.84 ± 0.020</td>
<td>0.84 ± 0.021</td>
<td>0.84 ± 0.022</td>
<td>0.84 ± 0.023</td>
</tr>
</tbody>
</table>

**Table 1. Evaluations on Willow Dataset.**

![Figure 3](image309x678 to 545x720) **Figure 3.** A random car example from Willow Object Classes [29].

![Figure 4](image13127) **Figure 4.** Samples from the quantum annealer can be helpful in improving the solution quality or reporting uncertainty.
the state of the art. Moreover, on a theoretical note, it enjoys polynomial speedup. Note that the exhaustive solution performs the best on this dataset. This further encourages research on making better quantum computers reducing the gap between the true and the machine-computed optima.

**Explaining the posterior.** While not being a true Bayesian inference, quantum annealing can provide samples from the energy landscape or the Hamiltonian. The samples could be used either, for instance, in associating a confidence to solutions or maybe in scenarios like active learning. To assess the usability of the samples, instead of looking into the lowest energy solution, we look at \( k \)-lowest energy solutions of the previous real data evaluation. We process them jointly and correct the erroneous bits by replacing them with the most frequent ones over all the samples. Fig. 4 shows on the real benchmark that for increasing \( k \) values, the accuracy also increases. This validates that different samples from a quantum annealer can be informative in exploring the energy landscape i.e., posterior defined in Sec. 4.

**Evaluations on the synthetic dataset.** We now interrogate various characteristics of our quantum approach, QuantumSync. Our results are plotted in Fig. 5. First, we inspect the behaviour under increasing noise. Fig. 5(a) shows that all the methods can handle low noise regimes. The increasing noise similarly impacts the methods we test. Our approach while not being superior to any, is on par. Fig. 5(b) shows that QuantumSync is significantly impacted from the increased problem sizes \((mn)^2\). This shows perhaps the most important limitation of current quantum computers, i.e., we can only reliably handle the problems of size \( < 64 \). Note that, this is also the size of our real sub-Willow dataset. The rest of the plots (c,d) show the impact of connectivity (graph completeness) on the solution quality. With noise \((\sigma = 0.1\) for this case) sparse graphs cause significant problems.

**Parameter selection.** We now evaluate, with the help of our synthetic data, how the solution and its probability changes w.r.t. the chain strength \( \chi \) and \( \lambda \). Both parameters strongly influence the probability to measure optimal solutions. \( \chi \) being too high keeps the chains unbroken during annealings, but adversely affects the solutions. Too low \( \chi \) lead to broken chains which, however, in many cases can be resolved with majority voting. Initial trial experiments help us to identify \( \chi = 3.0 \) and \( \lambda = 2.5 \) as optimal parameters which are kept fixed in our experiments. Note that the ablative study for \( \lambda \) on a classical computer in Sec. 5.2 also suggests \( \lambda > 2.0 \) as a suitable value for a range of \((n,m)\).

Tab. 2 highlights the effect of varying \( \chi \) in the tests with \( n = 3, m = 3 \) and \( n = 4, m = 4 \) on Advantage 1.1. We report the number of logical qubits of the target problem \( n_1 \), the number of physical qubits required to embed the problem on AQC \( n_{ph} \), the average maximum chain length of the embedding \( l \) and the average number of annealings leading to the optimal solution out of 200 samples. Each test for each \( \chi \) is repeated 50 times.

Table 2. The table summarises the effect of varying chain strength \( \chi \) on the number of measured optimal solutions out of 200, for \( n = 3, m = 3 \) (first row) and \( n = 4, m = 4 \) (second row).

<table>
<thead>
<tr>
<th>( n_1 / n_{ph} / l )</th>
<th>( \chi = 1 )</th>
<th>( \chi = 2 )</th>
<th>( \chi = 3 )</th>
<th>( \chi = 4 )</th>
<th>( \chi = 5 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( 18 / 48 / 3.2 )</td>
<td>126.9 ± 48.8</td>
<td>187.8 ± 11.5</td>
<td>180.9 ± 11.2</td>
<td>150.3 ± 19.9</td>
<td>85.3 ± 23.6</td>
</tr>
<tr>
<td>( 48 / 325 / 9.5 )</td>
<td>0.1 ± 0.30</td>
<td>9.9 ± 10.08</td>
<td>17.3 ± 14.0</td>
<td>6.1 ± 7.77</td>
<td>1.42 ± 2.72</td>
</tr>
</tbody>
</table>

**Different problem sizes and minor embedding.** Next, we systematically analyze which problem sizes can be successfully embedded on Pegasus topology of Advantage system 1.1 and solved globally optimally with high probability over 200 samples. In each configuration of \( n \) and \( m \) which can be successfully solved on D-Wave, we report the average number of measurements corresponding to the global optimum and its standard deviation over 50 repetitions with 200 annealings each. We also analyze minor embedding in the same experiments and report the average number of physical qubits used in the embedding, the maximum chain lengths along with their standard deviations over 50 runs. Fig. 6 visualises the experimental outcomes. We see that with the increasing number of logical qubits, the number of physical qubits required for the embedding as well as the maximum chain length increase (Fig. 6(a),(b)). For \( n = 3 \) and \( m = 3 \), the ratio is \( c = n_{ph} / n_1 \approx 2.65 \). It increases to \( c \approx 14.5 \) for \( n = 5 \) and \( m = 5 \). This is not surprising, since longer qubit chains increase the probability of chain
shows that: (i) for low noise regime, optimization indeed allows a QUBO-solver to achieve the global optimum for the problem at hand.

Hence, we investigate its behavior. Fig. 7 shows that over those seven experiments where \(\sigma = 0.2\), while the lowest energies between the two solutions can differ (a and b), for a wide variety of \(\lambda\)-choices the accuracy attained by binary optimization and permutation optimization can be very comparable (c vs d). As long as \(\lambda\) is not small (e.g., < 2), we observe almost identical performance. This positive result has motivated us to settle for a single value \(\lambda = 2.5\) for all of our evaluations (including Sec. 5.1).

Are permutation constraints effective? As D-Wave cannot search over the permutations but only over binary variables, it is of interest to see whether our permutation-ness regularization really works. To investigate that, we design random experiments \((n = 3, m = 3)\) with increasing noise (swap ratio) where the GT is known. We then form the constrained Q matrix and solve it via exhaustive search on a classical computer. Averaged over seven experiments, Fig. 8 shows that: (i) for low noise regime, optimizing over general binary variables \(B\) or over permutations \(P\) are indifferent and global optimum can always be found, (ii) for higher noise levels, while the energies attained seem to differ, the final accuracy is very similar. Hence, we conclude that injecting permutation constraints into Q as proposed is useful and makes it possible to use binary variables instead of permutations. This justifies why an adiabatic computer such as D-Wave could obtain global optima.

6. Conclusion

We presented QuantumSync, the first quantum approach to synchronization. We specifically focused on the group of permutations and showed how to formulate such problems for an adiabatic computer. We then used the cutting-edge quantum hardware to solve real-world problems with global guarantees. Our forward-looking experiments demonstrate that quantum computing hardware has reached the level that it can be applied to real-world problems with high potential to improve upon the known classical methods. We believe that our technique can inspire new generations of better algorithms for related and other computer vision problems, and we expect to see more work in the field in near future.

Acknowledgements. The work was supported by the ERC Grant 4DReply (770784), a Vannevar Bush Faculty fellowship, a grant from the Stanford-Ford Alliance and gifts from Amazon AWS and Snap, Inc.
References


