

# Quantum Permutation Synchronization

## —Supplementary Material—

Tolga Birdal<sup>1,\*</sup>

Vladislav Golyanik<sup>2,\*</sup>

Christian Theobalt<sup>2</sup>

Leonidas Guibas<sup>1</sup>

<sup>1</sup>Stanford University

<sup>2</sup>Max Planck Institute for Informatics, SIC

This part supplements our main paper by providing (i) proofs of the theorems and propositions contained in the main paper; (ii) further insights into the constraints that we propose; (iii) visualizations of minor embeddings on D-Wave; (iv) details and illustrations of the synthetic data; and (v) further descriptions of our real dataset. The figures and tables introduced in this document are referenced using Roman numerals, whereas the original references to the main matter are preserved.

### A. Proof of Theorems

*Proof of Theorem 1.* The proof follows from plugging Eq. (2) into the definition of a null-cycle in Eq. (1) of the main paper, obtaining:

$$f_c = f_{1,2} \circ f_{2,3} \circ \cdots \circ f_{(n-1),n} \circ f_{n,1} \quad (1)$$

$$= (f_1 \circ f_2^{-1}) \circ (f_2 \circ f_3^{-1}) \circ \cdots \circ (f_n \circ f_1^{-1}) \quad (2)$$

$$= f_1 \circ (f_2^{-1} \circ f_2) \circ (f_3^{-1} \circ f_3) \circ \cdots \circ (f_n^{-1} \circ f_n) \circ f_1^{-1} \quad (3)$$

$$= f_1 \circ f_1^{-1} = f_{null} \quad \forall c \in \mathcal{C}. \quad (4)$$

□

*Proof of Theorem 2.* Without loss of generality, for an edge  $(i, j)$ , we consider the transformed cycle consistency constraint:

$$(f_i \circ f_j) \circ (f_j \circ f_i)^{-1} = (f_i \circ f_j) \circ (f_j^{-1} \circ f_i^{-1}) = f_i \circ f_i^{-1}. \quad (5)$$

This shows that consistency relation is unchanged under the action of an arbitrary element in the group  $f_g$ . □

### B. Extended Proof of Proposition 1

Here we provide more steps to the derivation in Prop. 1 of the main paper:

$$\mathbf{X}^* = \arg \min_{\{\mathbf{P}_i \in \mathcal{P}_n^N\}} \sum_{(i,j) \in \mathcal{E}} \|\mathbf{P}_{ij} - \mathbf{P}_i \mathbf{P}_j^\top\|_{\mathbb{F}}^2 \quad (6)$$

$$= \arg \min_{\{\mathbf{P}_i \in \mathcal{P}_n^N\}} \sum_{(i,j) \in \mathcal{E}} \|\mathbf{P}_{ij}\|_{\mathbb{F}}^2 - \|\mathbf{P}_i \mathbf{P}_j^\top\|_{\mathbb{F}}^2 - 2\text{tr}(\mathbf{P}_j \mathbf{P}_i^\top \mathbf{P}_{ij})$$

$$= \arg \min_{\{\mathbf{P}_i \in \mathcal{P}_n^N\}} 2N^2n - 2 \sum_{(i,j) \in \mathcal{E}} \text{tr}(\mathbf{P}_j \mathbf{P}_i^\top \mathbf{P}_{ij}) \quad (7)$$

$$= \arg \min_{\{\mathbf{P}_i \in \mathcal{P}_n^N\}} - \sum_{(i,j) \in \mathcal{E}} \text{tr}(\mathbf{P}_j \mathbf{P}_i^\top \mathbf{P}_{ij}) \quad (8)$$

$$= \arg \min_{\{\mathbf{P}_i \in \mathcal{P}_n^N\}} - \sum_{(i,j) \in \mathcal{E}} \text{vec}(\mathbf{P}_i)^\top (\mathbf{I} \otimes \mathbf{P}_{ij}) \text{vec}(\mathbf{P}_j)$$

$$= \arg \min_{\{\mathbf{P}_i \in \mathcal{P}_n^N\}} - \sum_{(i,j) \in \mathcal{E}} \mathbf{q}_i^\top (\mathbf{I} \otimes \mathbf{P}_{ij}) \mathbf{q}_j = \arg \min_{\{\mathbf{P}_i \in \mathcal{P}_n^N\}} \mathbf{q}^\top \mathbf{Q}' \mathbf{q}. \quad (9)$$

---

\* both authors contributed equally to this work

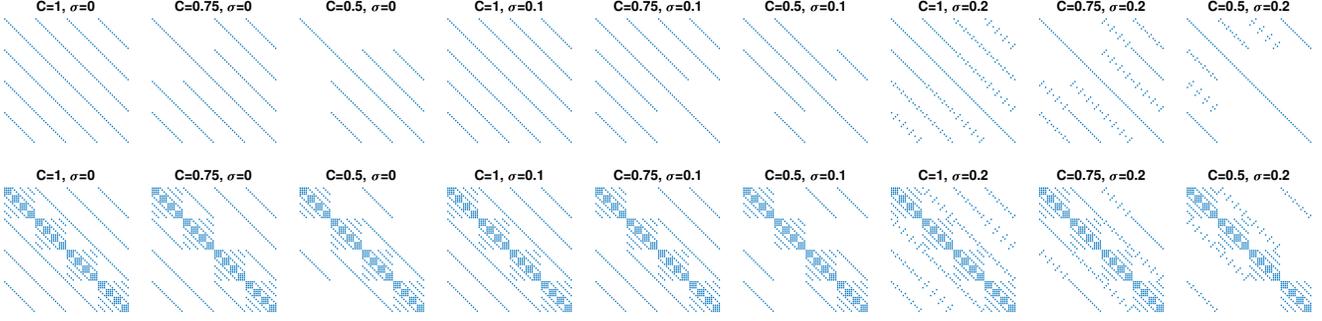


Figure I. Sparsity patterns of the unconstrained (**top row**) and constrained (**bottom row**)  $\mathbf{Q}$ -matrices. For the former we use  $\lambda = 0$  while the latter uses  $\lambda = 2$ . The matrix is constructed for  $n = m = 4$  and thus has  $mn^2 = 64$  elements per dimension. We visualize the matrices for a range of completeness  $C$  and swap-ratio  $\sigma$  values.

### C. Proof of Proposition 2

Linear constraints can be injected into a QUBO problem in the following manner:

$$\mathbf{x}^* = \arg \min_{\mathbf{x} \in \mathcal{B}} \mathbf{x}^\top \mathbf{Q}' \mathbf{x} + \lambda \|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2^2 \quad (10)$$

$$= \arg \min_{\mathbf{x} \in \mathcal{B}} \mathbf{x}^\top \mathbf{Q}' \mathbf{x} + \lambda (\mathbf{A}\mathbf{x} - \mathbf{b})^\top (\mathbf{A}\mathbf{x} - \mathbf{b}) \quad (11)$$

$$= \arg \min_{\mathbf{x} \in \mathcal{B}} \mathbf{x}^\top (\mathbf{Q}' + \lambda \mathbf{A}^\top \mathbf{A}) \mathbf{x} - 2\lambda \mathbf{b}^\top \mathbf{A} \mathbf{x} \quad (12)$$

$$= \arg \min_{\mathbf{x} \in \mathcal{B}} \mathbf{x}^\top \mathbf{Q} \mathbf{x} + \mathbf{s}^\top \mathbf{x}. \quad (13)$$

### D. Example Constraint Matrices and Sparsity Patterns

We have shown in the main paper that the permutation constraints can be formulated as a set of linear systems  $\{\mathbf{A}_i \mathbf{q}_i = \mathbf{b} \triangleq \mathbf{1}\}$ . We now show concrete examples in two and three dimensions of these constraint matrices:

1.  $n = 2$ :

$$\mathbf{A}_i = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix}, \quad \mathbf{b}_i = \mathbf{1} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}. \quad (14)$$

2.  $n = 3$ :

$$\mathbf{A}_i = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \end{bmatrix}, \quad \mathbf{b}_i = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}. \quad (15)$$

We further analyze how our quadratic constraint matrix looks like when these linear constraints are added. For different random experiments, we plot in Fig. I these  $\mathbf{Q}$ -matrices for different values of completeness  $C$  and swap ratio  $\sigma$  when  $n = m = 4$ . Note that this is a typical setting for our synthetic evaluations.

### E. Minor Embeddings and Sampling

In the default mode, *Leap2* allocates couplings between qubits in the minor embeddings even if they are equal to zero. If zero couplings are explicitly avoided, the total number of physical qubits required to minor-embed a logical problem, along with the maximum required chain length, decreases, and the probability of measuring the optimal solution in single annealing is increasing. See Fig. II for the variant of Fig. 6 of the main paper, without allocated zero qubit couplings. We observe empirically that more compact minor embeddings—in contrast to what one could presume—do not allow to solve larger problem instances.

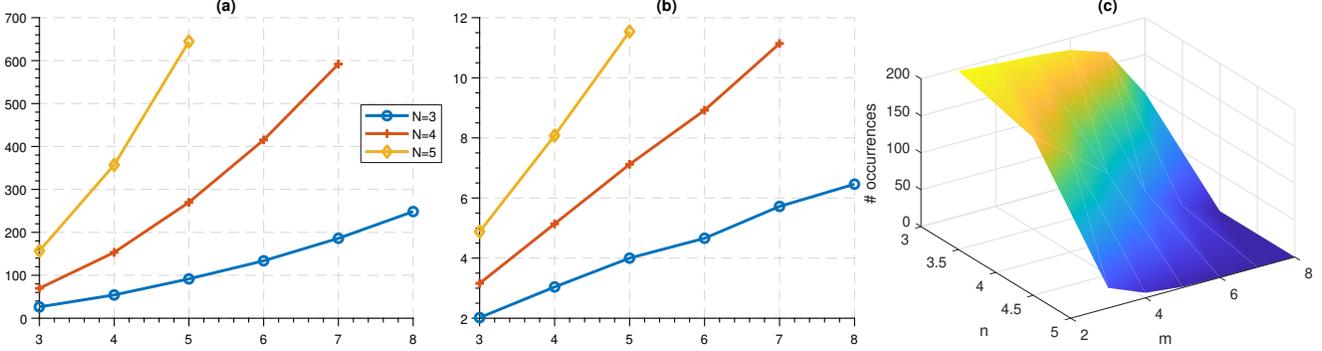


Figure II. This figure supplements Fig. 6 of the main paper with the difference that the zero qubit couplings are excluded. For different  $n$  and an increasing number of views  $m$ , (a) plots the number of qubits required to map a problem; and (b) at  $\chi = 3.0$ , shows the required maximum chain length required to embed the problem on Advantage 1.1. (c) plots the average number of measured optimal solutions in 200 samples, for different pairs of  $n$  and  $m$  (averaged over 50 repetitions).

Table I. The table summarises the average number (and the corresponding standard deviation) of measured optimal solutions out of 200, for  $n = 4, m = 4$  and varying chain strength  $\chi$ , strictly as derived in Prop. 2 (first row) and with adjusted weights as described in Sec. E, *i.e.*,  $-\sqrt{|s_k|}$  (second row). Each number is reported for 50 repetitions of the experiment.

	$\xi = 1$	$\xi = 2$	$\xi = 3$	$\xi = 4$	$\xi = 5$
derived	$0.02 \pm 0.14$	$1.26 \pm 2.1$	$1.38 \pm 1.75$	$0.08 \pm 0.27$	$0.0 \pm 0.0$
adjusted	$0.1 \pm 0.30$	$9.9 \pm 10.08$	$17.3 \pm 14.0$	$6.1 \pm 7.77$	$1.42 \pm 2.72$

In Fig. III, we show three exemplary minor embeddings on the Pegasus architecture of Advantage system 1.1 for problems of different sizes. While the number of logical qubits for the shown embeddings ranges from 18 (A) to 63 (C), the number of the corresponding physical qubits ranges from 49 (A) to 550 (C). Note that the shown visualizations include zero couplings, and more compact minor embeddings are possible (*cf.* Fig. II). With the increasing problem size, we also see that maximum lengths of qubit chains  $l$  in the embeddings, which encode the same logical qubit, increase as well. While  $l = 4$  for  $n = 3, m = 4$ , it increases to 10 and 13 for the combinations  $n = 4, m = 4$  and  $n = 3, m = 8$ , respectively. For the same problems, AQC of the previous generation 2000Q requires 108, 782 or 1378 physical qubits on average over 50 runs, respectively.

**Selecting qubit biases.** We empirically find that the qubit biases have to be set differently compared to as derived in (15), see Tab. I for the summary of our experiment. Thus,  $-\sqrt{|s_k|}$  worked well (instead of the derived  $-\text{diag}(\mathbf{Q})_k - s_k$ )<sup>1</sup> in combination with the selected chain strengths and  $\lambda$ , see Sec. 5.1 for details on parameter selection. The possible reasons for that lie on the hardware side. First, the qubit biases and couplings are converted to magnetic fields acting on qubits, *i.e.*, the weight encoding and annealing are analogue physical processes. The embedded problem is predominantly defined in terms of qubit couplings ( $\frac{k^2-k}{2}$  logical couplings at most in contrast to  $k$  logical biases at most for a problem with  $k$  logical qubits). Moreover, magnetic fields are imposed to keep chains of physical qubits intact. Last but not least, the range of real (floating-point) values which can be mapped to the native Ising format of D-Wave is limited (currently, it is  $[-2; 2]$  for biases and  $[-1; 1]$  for couplings on 2000Q and system 1.1), and analytically derived biases and couplings have to be scaled down to the supported ranges. These factors can lead to offsets in qubit biases which are difficult to predict theoretically, as QUBO formulations are often derived without consideration of the minor embedding to a real AQC.

**On sampling.** Note that, theoretically the samples coming from the quantum annealer can not be directly interpreted as the samples from the induced Boltzmann distribution characterized by  $\beta$  (*e.g.*, lower values of  $\beta$  result in samples less constrained to the lowest energy states), as the annealer samples a modified posterior. Optionally, one could steer the samples towards the local minima using a CPU-based descent algorithm. However, we found that in practice this hack, also suggested by D-Wave, does not work well for capturing a diverse set of modes. Nevertheless, as we show in Fig. 4 (in the main paper) using the original samples as alternative plausible solutions could still boost the accuracy albeit incrementally. While a true posterior adjustment is not yet implemented in D-Wave, we foresee that upcoming years would witness a leap on these fronts.

<sup>1</sup>Our experiments suggest that there is a broader range of smaller biases which work as well as  $-\sqrt{|s_k|}$ . In several our test cases,  $-0.5s_k$  were leading to similar solution distributions. Further study is required on the differences between the derived weights and the weights which should be set in practice.

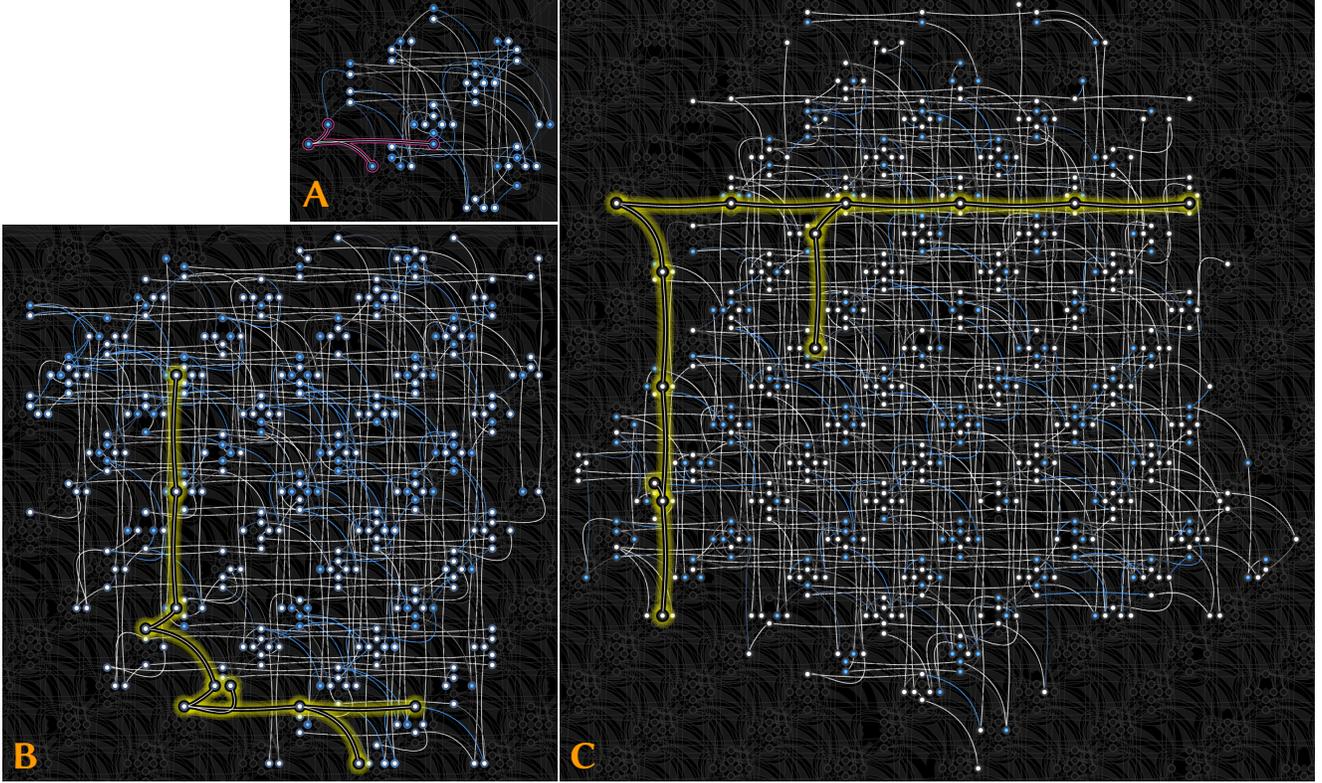


Figure III. Exemplary minor embeddings in the experiments with  $n = 3, m = 3$  (A,  $n_{ph} = 49$ ),  $n = 4, m = 4$  (B,  $n_{ph} = 341$ ) and  $n = 3, m = 8$  (C,  $n_{ph} = 550$ ). In each case, we highlight qubit chains of the maximum chain length in the embedding, either in magenta (A, no warnings) or yellow (B and C, chain length warnings). Note that the shown minor embeddings include zero qubit couplings.

## F. Further Details on Synthetic Experiments

We visualize in Fig. IV some examples from our random synthetic dataset. While the common case of  $n = 4$  and  $m = 4$  can be visualized this way, the same does not hold true for other  $n, m$  combinations. That is why the points on the grid only assist the visualization, and are not used in practice. The important cues are the correspondences denoting permutations, whose rows might be randomly swapped to inject noise.

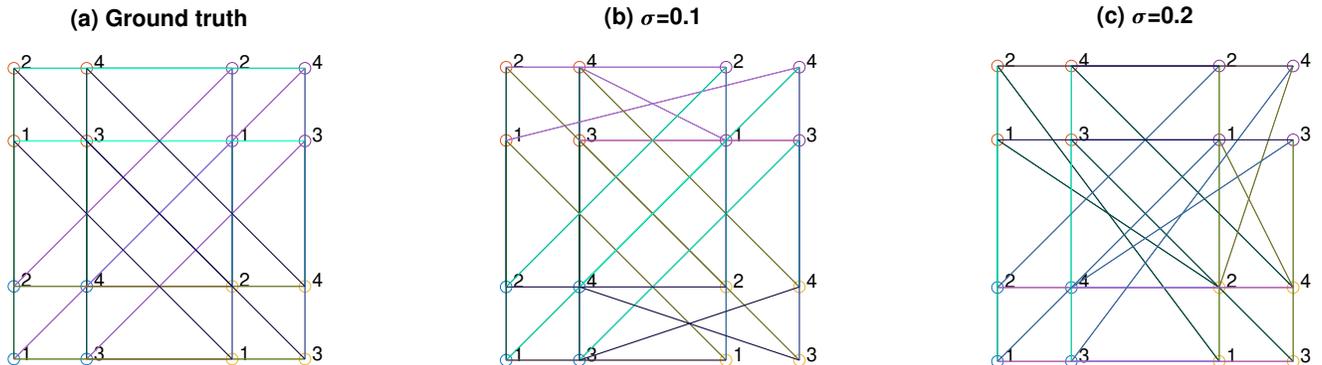


Figure IV. Samples from our fully connected synthetic dataset for different values of swap ratio  $\sigma$ . For sparser graphs, where completeness is less than one, it is possible to imagine edges being dropped from these graphs. In the figure, each group (indicated using differently colored points) corresponds to a *view* and each inter-group correspondence corresponds to a permutation that we optimize for. Note that the points are drawn as a grid to ease visual perception. Neither our algorithm nor the state-of-the-art methods we compare would use this information.

## G. Evaluations on the Real Dataset Based on the Willow Object Classes [1]

**Dataset description.** Willow dataset is composed of 40 RGB images of four object classes: *duck*, *car*, *winebottle*, *motorbike* and *face*. These images are extracted either from *Caltech-256* or *PASCAL VOC* datasets. The *in-the-wild* nature of these images as shown in Figs. V and VII makes simple template matching ill-suited for correspondence estimation. This is why, for our evaluations—as explained in the main paper—we followed the same design as Wang *et al.* [7] and benefited from the state-of-the-art-deep learning approaches. The authors of [1] have manually selected ten distinctive points on the target object (category). These keypoints are annotated consistently across all the instances in each of the categories as we show in Fig. V. To this end, a semi-automatic graph matching has been used as explained in [1]. In our paper, we have further extracted 35 subsets of this dataset composed of only the first four annotations ( $n = 4$ , as shown in Fig. V) and four images which were consecutive ( $m = 4$ ). Thus, we have 35 subset graphs. Due to our automated initial graph matching process, while the keypoint locations are always correct, the permutation matrices  $\mathbf{P}_{ij}$  are contaminated with real noise (Fig. VIII).



Figure V. A random example from the *cars* class of Willow Object Classes dataset [1].

**Extended plots of the evaluations.** In the paper we have reported the average accuracy over all the images in our modified-willow dataset. However, it is also of interest to see how our algorithm performs on individual subsets. We plot in Fig. VI the performance of all the approaches under test, over all the individual subgraphs. It is visible that all the methods perform similarly. While our algorithm has an overall advantage, it can fail on certain examples, despite the quest for the global minimum. This explains the slightly lower accuracy with respect to the exhaustive solution.

**Evaluation metrics.** In the main paper, we mention that we report the number of bits correctly detected. This so-called *accuracy* measure between the estimated permutations  $\mathbf{X}$  and their ground truth  $\bar{\mathbf{X}}$  is formally defined as:

$$\epsilon(\mathbf{X}, \bar{\mathbf{X}}) = 1 - \frac{1}{mn^2} \sum_{i=1}^m |\text{vec}(\mathbf{X}_i \oplus \bar{\mathbf{X}}_i)|_1, \quad (16)$$

where  $\oplus$  is the *exclusive-or* operand. Intuitively, this is a Hamming similarity derived from the Hamming distance in which an error is made whenever two bits differ.

## H. Detailed Analysis of Several Recent Works on Quantum Computer Vision

Algorithm	Problem	Processor	# qubits	total QPU time
QUBO Suppression [5]	non-maximum suppression in human tracking	2X	1000	n/a
QA [2, 3]	transformation estimation and point set alignment	2000Q	2048	60 sec.*
QGM [6]	graph matching (two graphs, up to four points)	2000Q	2048	2 – 2.5 min.
<i>QuantumSync</i> (ours)	permutation synchronization (multiple views, multiple points)	Adv. 1.1	5436	> 15 min.

Table II. Overview of several recent quantum computer vision methods published at computer vision conferences and our *QuantumSync*. Note that the right-most column reports the overall experimental QPU runtime in the evaluation of the methods. “\*”: QA has been recently tested on D-Wave 2000Q; the results are reported in the supplementary document [3].

In this section, we analyze several recent works on quantum computer vision [2, 5, 6] and *QuantumSync*. We summarize the problems, the D-Wave processors and the experimental QPU time used in the experiments on real quantum hardware in Tab. II. The QUBO suppression approach [5] solves an existing QUBO formulation of non-maximum suppression on

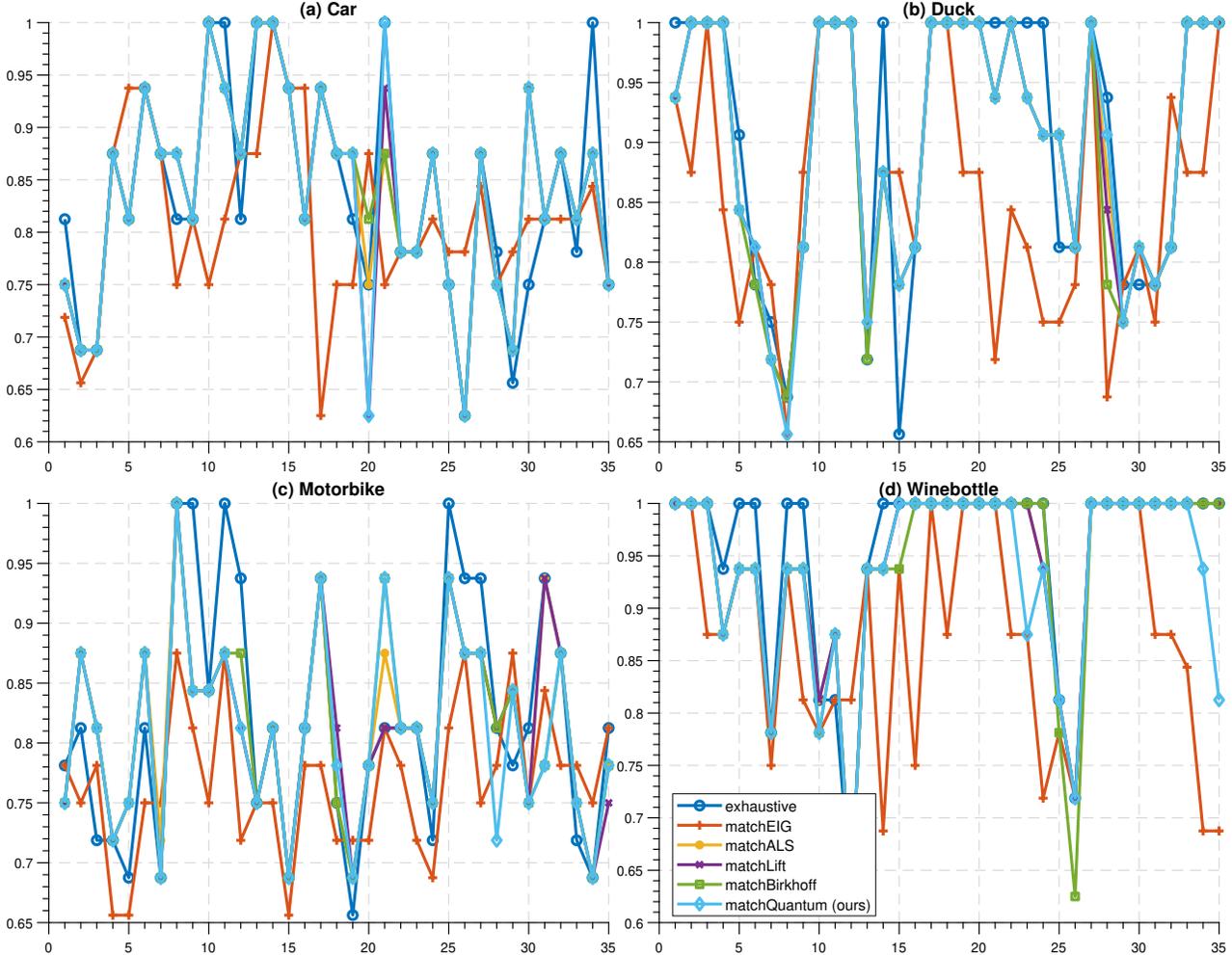


Figure VI. Detailed evaluations over all the subsets on the Willow Object Classes dataset [1].

D-Wave 2X with 1000 qubits. The experimental results show improvements in solving the target combinatorial optimization problem on quantum hardware, and the authors conclude that the age of quantum computing for human tracking has arrived.

Quantum Approach (QA) to correspondence problems on point sets, inspired by the altered gravitational model for point set alignment [4], was proposed in [2], where the cases with and without known correspondences between the points have been considered. In QA, final rigid transformations are approximated as a linear combination of basis elements which, in the general case, allows representing affine transforms. Thus, the method can be extended to affine transforms in a straightforward way. QA was confirmed on D-Wave 2000Q as reported in the supplementary material accompanying the paper [3]. QA can align two point sets at a time. Even though the dimensionality of the resulting matrix of couplings and biases does not depend on the number of points (the size of this matrix depends on the cardinality of the basis elements set), the super-linear complexity to prepare the state for large problems on classical hardware can result in large runtimes.

The method of Benkner *et al.* [6] for quantum graph matching (QGM) is concurrent to our work. While they propose a similar mechanism to impose permutation matrix constraints, there are multiple fundamental differences. First, QGM is designed for operation on two graphs, whereas we propose a permutation (map) synchronization algorithm for multiple views/scans with multiple points each. In our problem setting, initial (noisy) estimates of the pairwise permutations are given. Second, we successfully confirm our method for up to nine views with three points each, or seven views with four points each, whereas in [6], only the  $3 \times 3$  case has been successfully solved on a real AQC. Another distinguishing aspect of our work is that we achieve the probability of over 90% to measure a correct optimal solution for the permutation synchronization problem for the  $3 \times 3$  case, whereas [6] this probability (for a different problem though) only slightly exceeds the probability of randomly guessing a correct  $3 \times 3$  permutation (*i.e.*,  $\approx 16.7\%$ ). While the experiments in [6] are performed on 2000Q as



Figure VII. Initial matches on random examples from Willow Object Classes dataset [1].

opposed to the Advantage system 1.1 (the latest generation of D-Wave AQC) we use, the difference in the probabilities is unlikely to stem from the differences between the hardware architectures. We have tested our algorithm on 2000Q as well and even observed slightly higher probabilities for the  $3 \times 3$  case, though larger problems either resulted in lower probabilities compared to Advantage system 1.1 or could not be embedded on the 2000Q QPU and solved on it at all.

## References

- [1] Minsu Cho, Karteek Alahari, and Jean Ponce. Learning graphs to match. In *International Conference on Computer Vision (ICCV)*, pages 25–32, 2013. [5](#), [6](#), [7](#)
- [2] Vladislav Golyanik and Christian Theobalt. A quantum computational approach to correspondence problems on point sets. In *Computer Vision and Pattern Recognition (CVPR)*, 2020. [5](#), [6](#)
- [3] Vladislav Golyanik and Christian Theobalt. A quantum computational approach to correspondence problems on point sets (supplementary material about the experiments on d-wave 2000q). In *Computer Vision and Pattern Recognition (CVPR)*, 2020. [5](#), [6](#)
- [4] Vladislav Golyanik, Christian Theobalt, and Didier Stricker. Accelerated gravitational point set alignment with altered physical laws. In *International Conference on Computer Vision (ICCV)*, 2019. [6](#)
- [5] Junde Li and Swaroop Ghosh. Quantum-soft qubo suppression for accurate object detection. In *European Conference on Computer Vision (ECCV)*, 2020. [5](#)
- [6] Marcel Seelbach Benkner, Vladislav Golyanik, Christian Theobalt, and Michael Moeller. Adiabatic quantum graph matching with permutation matrix constraints. In *International Conference on 3D Vision (3DV)*, 2020. [5](#), [6](#)
- [7] Qianqian Wang, Xiaowei Zhou, and Kostas Daniilidis. Multi-image semantic matching by mining consistent features. In *Computer Vision and Pattern Recognition (CVPR)*, 2018. [5](#)