Quantum Permutation Synchronization

Stanford University
max planck institut informatik

Tolga Birdal **Vladislav Golyanik** Christian Theobalt Leonidas Guibas

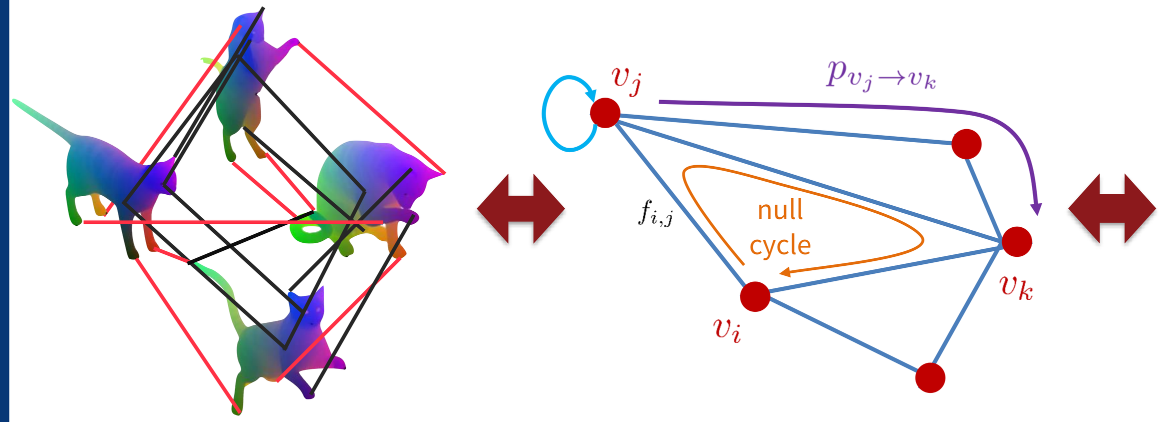
http://quantumcomputervision.github.io/

CVPR VIRTUAL JUNE 19-25

Introduction

Our Problem: Permutation Synchronization

Matching not just two, but *n* different sets of objects to each other, **jointly** [1]. In other words, a *multi-way matching*. In the scenario where correspondences are bijective, the problem converts to ensuring *cycle consistency* in the graph of **permutations** [2]:



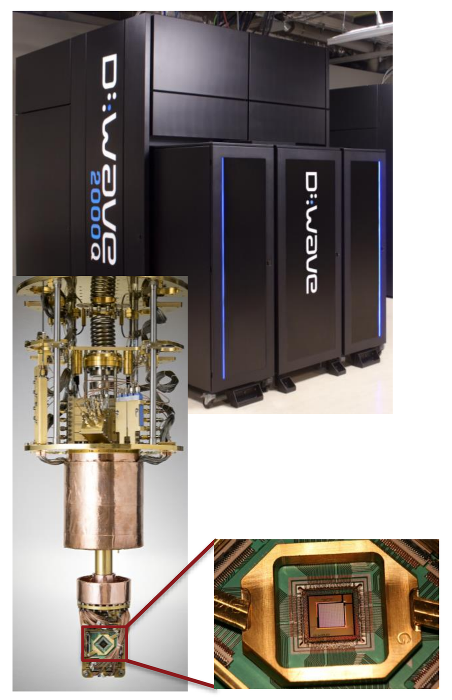
$$\arg\min_{\mathbf{P}\in\mathcal{P}_n^N} \sum_{(i,j)\in E} \|\mathbf{P}_{ij} - \mathbf{P}_i\mathbf{P}_j^\top\|_F^2$$

Given Unknown

Multiple graph matching — Ensuring all cycles are *null* — Minimizing *cycle consistency constraint*

We solve the **non-convex**, **combinatorial** permutation synchronization problem **without relaxation** on a **real quantum computer.**

Adiabatic Quantum Computer Vision (AQC-V)



dwavesys.com/resources /media-resources

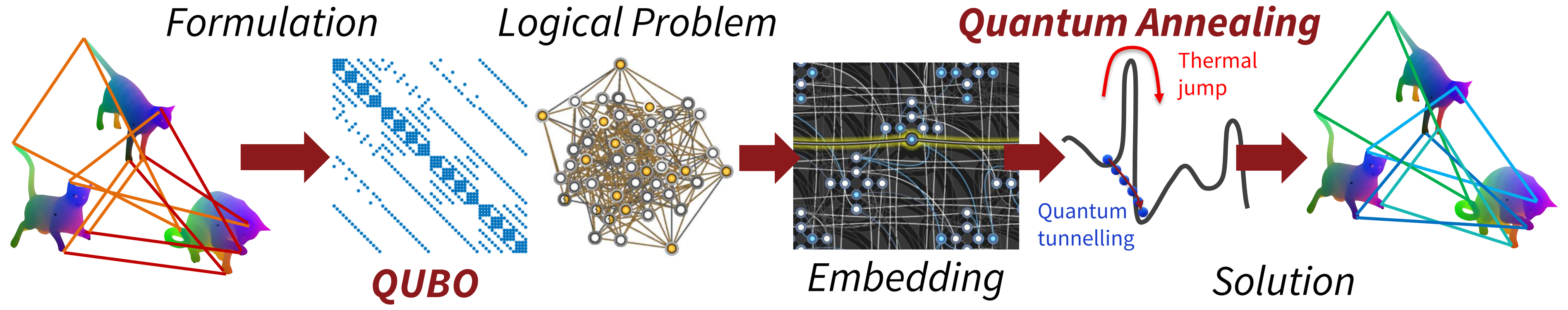
	QUBO Suppression ECCV'20	Quantum Alignment CVPR'20	Quantum Graph Matching 3DV'2020	**Quantum Sync This Paper**
Quantum Processor	D-Wave 2X 1000 qubits	D-Wave 2000Q 2048 qubits	D-Wave 2000Q 2048 qubits	D-Wave Advantage 1.1 5436 qubits
AQC Problem	Non-maximum suppression	Pairwise point set alignment	Pairwise graph matching	Permutation synchronization
QPU Time	N/A	1 minute	2-3 minutes	>15 minutes

Contributions

(**a**) Formulating a **QUBO** for permutation synchronization with **permutation-ness** as a **linear** constraint

(**b**) Extensive evaluation on a **real Quantum Computer, D-Wave Advantage 1.1**

Our Approach: QuantumSync



1. Formulating the Vanilla QUBO

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

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

QUBO

where $\mathbf{x}_i = \mathrm{vec}(\mathbf{X}_i)$, $\mathbf{x} = [\cdots \mathbf{x}_i^\top \cdots]^\top$ *and:*

$$\mathbf{Q}' = -\begin{bmatrix} \mathbf{I}\otimes\mathbf{P}_{11} & \mathbf{I}\otimes\mathbf{P}_{12} & \cdots & \mathbf{I}\otimes\mathbf{P}_{1m} \\ \mathbf{I}\otimes\mathbf{P}_{21} & \mathbf{I}\otimes\mathbf{P}_{22} & \cdots & \mathbf{I}\otimes\mathbf{P}_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{I}\otimes\mathbf{P}_{m1} & \mathbf{I}\otimes\mathbf{P}_{m2} & \cdots & \mathbf{I}\otimes\mathbf{P}_{mm} \end{bmatrix}.$$

Q is **symmetric** and no other constraints are needed.

2. Permutations as Linear Constraints

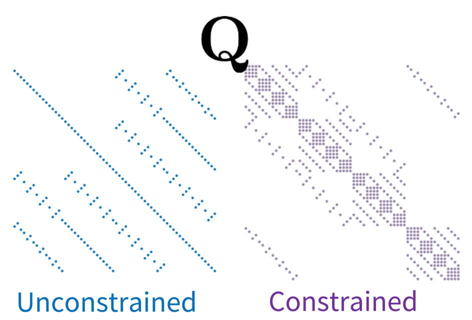
Group of permutations : $\mathcal{P}_n := \{\mathbf{P}\in\{0,1\}^{n\times n} : \mathbf{P}\mathbf{1}_n = \mathbf{1}_n,\ \mathbf{1}_n^\top\mathbf{P} = \mathbf{1}_n^\top\}$

Binary — Rows sum to 1 — Cols sum to 1

Hence, for all variables we would like to solve for : $\mathrm{diag}(\mathbf{A}_1,\cdots,\mathbf{A}_n)\mathbf{x} = \mathbf{1}$, $\quad \mathbf{A}_i = \begin{bmatrix}\mathbf{I}\otimes\mathbf{1}^\top \\ \mathbf{1}^\top\otimes\mathbf{I}\end{bmatrix}$

3. Incorporating Linear Constraints into QUBO

Proposition 2. *The constrained problem can be formulated into an unconstrained QUBO:*



$$\arg\min_{\mathbf{x}\in\mathcal{B}^n} \mathbf{x}^\top\mathbf{Q}\mathbf{x} + \mathbf{s}^\top\mathbf{x}$$

Binary Variables — Quadratic Term — Linear Term

$$\mathbf{Q} = \mathbf{Q}' + \lambda\mathbf{A}^\top\mathbf{A}$$

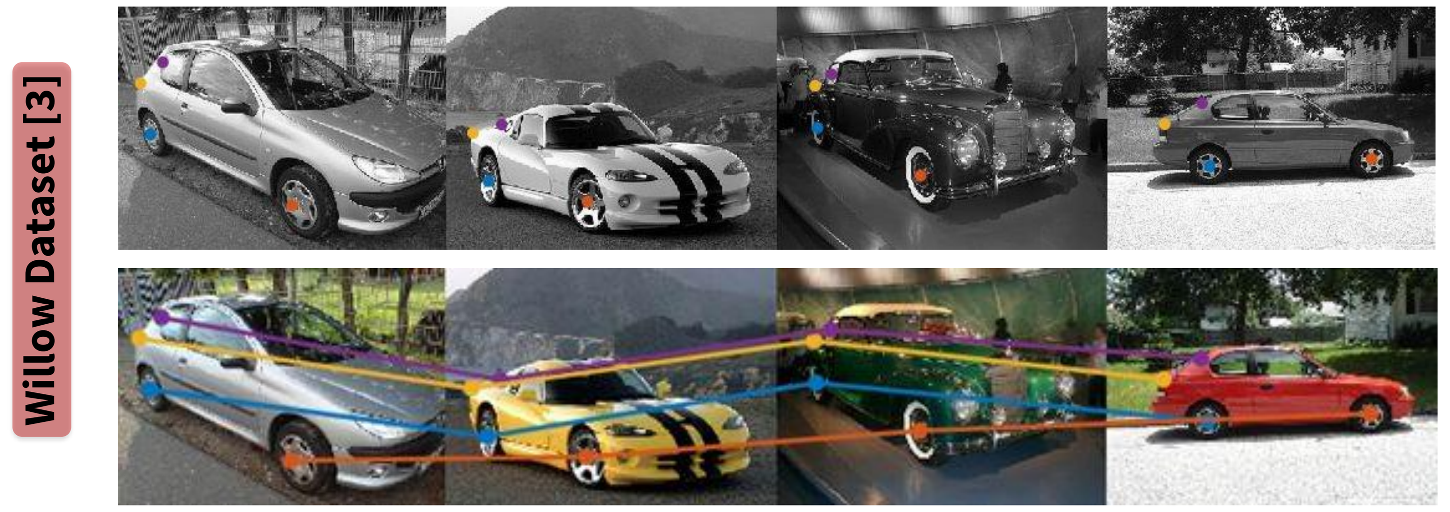
$$\mathbf{s} = -2\lambda\mathbf{A}^\top\mathbf{b}$$

Experimental Evaluation

1. Solving Real Problems on D-Wave Advantage 1.1

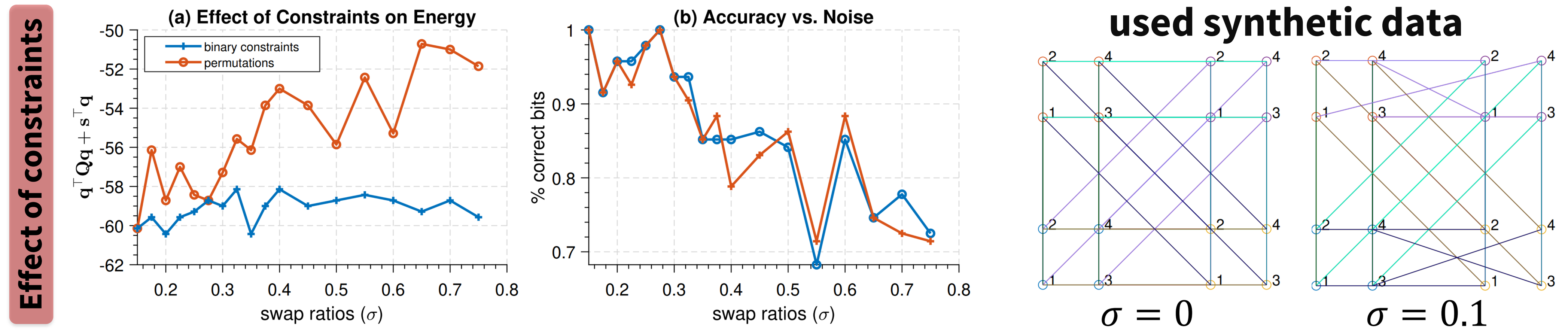
D-Wave Python API: *docs.ocean.dwavesys.com/en/stable/*

We can match the state-of-the-art methods in small problems (*n*=4, *m*=4).



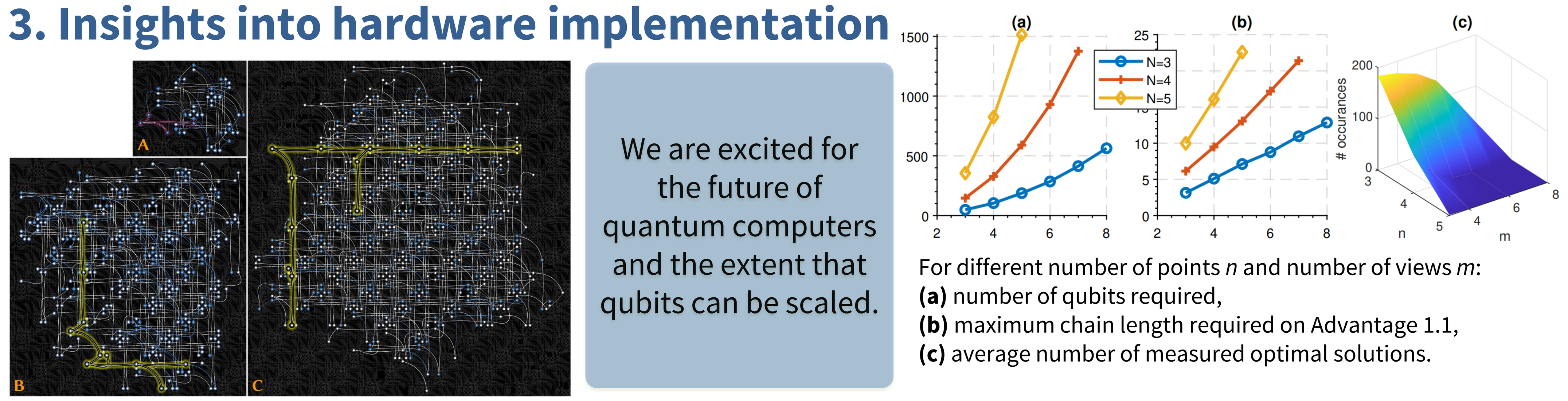
	Average
Exhaustive	**0.88 ± 0.104**
EIG	0.83 ± 0.088
ALS	0.87 ± 0.092
LIFT	0.87 ± 0.094
Birkhoff	0.87 ± 0.093
D-Wave(Ours)	0.87 ± 0.096

2. Impact of regularization (Binary variables vs. Permutations)



Our constraint injection scheme yields valid permutations while maintaining the solution quality.

3. Insights into hardware implementation



We are excited for the future of quantum computers and the extent that qubits can be scaled.

For different number of points *n* and number of views *m*:
(a) number of qubits required,
(b) maximum chain length required on Advantage 1.1,
(c) average number of measured optimal solutions.

- Our forward-looking experiments demonstrate that quantum hardware has reached the level that it can be applied to real-world problems.
- We hope to inspire and foster new and exciting **research in quantum computer vision.**

References

1. Pachauri, Deepti, Risi Kondor, and Vikas Singh. "Solving the multi-way matching problem by permutation synchronization." *NeurIPS* 2013.
2. Birdal, Tolga, and Umut Simsekli. "Probabilistic permutation synchronization using the riemannian structure of the birkhoff polytope." *CVPR* 2019.
3. Cho, Minsu, Karteek Alahari, and Jean Ponce. "Learning graphs to match." *ICCV* 2013.