Solving Combinatorial Optimization Problems in Computer Vision via Quantum Annealing

DISSERTATION
zur Erlangung des Grades eines Doktors
der Naturwissenschaften

vorgelegt von

André Marcel Seelbach Benkner, M.Sc.

eingereicht bei der Naturwissenschaftlich-Technischen Fakultät
der Universität Siegen
Siegen 2025

Betreuer und erster Gutachter
Prof. Dr. Michael Möller
Universität Siegen

Zweiter Gutachter
Prof. Dr. Tolga Birdal
Imperial College London

Tag der mündlichen Prüfung
30.04.2026

Abstract

Since the ability of quantum computing to speed up certain, specific computations has a solid theoretical underpinning, it is a natural question to ask if problems in computer vision also benefit from it. This doctoral thesis is in particular concerned with the question if the recently emerging optimization method of quantum annealing is useful for solving combinatorial optimization problems that arise in computer vision. The computer vision problem that is mainly investigated here is the problem of shape matching. The task of shape matching is to obtain the correspondences of two objects in a scene that are described with point clouds. Most often, one is concerned with finding the correspondences for a person who is recorded in 3D two times but with a different pose. The reason why this computer vision problem was picked as an interesting candidate for an application of quantum annealing is because it can be formulated as a difficult combinatorial optimization problem with a quadratic objective. This is close to the type of optimization problem that recent quantum annealing devices try to solve. The biggest difference is that in quantum annealing one a priori has unconstrained problems so that one cannot include equality constraints in the optimization. In the first work of this thesis, multiple methods are presented to overcome this difference and for really small shape matching problems experiments with simulated quantum annealing and with quantum annealing on D-Wave quantum computers were presented. In the second work of this thesis an iterative method was developed so that it was even possible to solve shape matching problems with over 500 vertices. Finally, a third publication looked at problems where one needs to find the correspondences between multiple shapes.

Besides applications of quantum annealing for shape matching, this thesis also contains work on solving the motion segmentation problem with quantum annealing and work about integrating a quantum annealer in a neural network to solve computer vision problems. The integration of the quantum annealer in a neural network can also be viewed as the task to learn an appropriate formulation of a computer vision problem in a way that is solvable with a current quantum annealer. Finally, we also present work about an iterative method for compensating limited connectivity in quantum annealing devices. This work has no particular computer vision application in mind, but investigates the performance of the different approaches on Max-Cut problems.

The thesis is structured in a cumulative way, so that the individual publications are presented separately. On a broader picture, the works are motivated by the following questions: What could be the impact of quantum annealing on computer vision and how could computer vision algorithms that use quantum annealing look like? And from another perspective: How would the field of computer vision be changed if there were massive advances in building quadratic, unconstrained, binary optimization problem solvers with or even without utilizing quantum computing?

Zusammenfassung

Die Fähigkeit von Quantum Computing bei speziellen Problemen Berechnungen schneller als herkömmliche Computer durchzuführen, hat ein solides theoretisches Fundament. Aus diesem Grund stellt sich die Frage, ob Quantum Computing auch im Fachbereich der Computer Vision Vorteile verspricht. Die vorliegende Doktorarbeit beschäftigt sich damit, ob Optimierungsmethoden wie Quantum Annealing nützlich sind, um kombinatorische Optimierungsprobleme aus dem Bereich der Computer Vision zu lösen. Das Optimierungsproblem, das hier am meisten untersucht wurde, heißt Shape Matching. Das Ziel von Shape Matching ist es, Korrespondenzen zwischen zwei mit Punktwolken dargestellten Objekten einer Szene zu finden. Meistens sollen die Korrespondenzen zwischen zwei unterschiedlichen 3D Aufnahmen einer Person bestimmt werden, wobei sich die Pose geändert hat. Ein Hauptgrund, wieso gerade dieses Optimierungsproblem interessant für diese Untersuchung ist, ist die Möglichkeit es als kombinatorisches Optimierungsproblem mit quadratischen Kosten zu formulieren. Dies ist sehr ähnlich zu den Optimierungsproblemen, die durch Quantum Annealing zurzeit gelöst werden können. Der größte Unterschied ist, dass bei Quantum Annealing a priori nur unrestringierte Probleme gelöst werden können. Also ist es zunächst nicht möglich lineare Gleichheitsbedingungen in der Optimierung zu berücksichtigen. In der ersten Veröffentlichung in dieser Doktorarbeit wurden mehrere Methoden vorgestellt, um diese Unterschiede in den Optimierungsproblemen zu überwinden. Auch wurden Experimente für kleine Shape Matching Instanzen mittels simulierten Quantum Annealing und Quantum Annealing auf D-Wave Quantencomputern durchgeführt. Bei der zweiten Publikation wurde eine iterative Methode entwickelt, welche es ermöglicht Shape Matching Probleme mit mehr als 500 3D-Punkten zu lösen. Schließlich betrachtet eine dritte Publikation das Problem, dass Korrespondenzen zwischen mehreren 3D-Shapes gleichzeitig mit Quantum Annealing ermittelt werden sollen.

Neben den Anwendungen von Quantum Annealing für Shape Matching enthält die Doktorarbeit auch Arbeiten darüber, wie Quantum Annealing für Motion Segmentation verwendet werden kann. Außerdem gibt es eine Veröffentlichung darüber, wie man einen Quantum Annealer in ein neuronales Netz integrieren kann, um Computer Vision Probleme zu lösen. Dies kann auch als ein Lernverfahren interpretiert werden, um das Aufstellen des Optimierungsproblems, welches für den Quantum Annealer geeignet ist, aus dem Computer Vision Problem heraus zu automatisieren. Weiterhin wird auch eine Arbeit behandelt, bei der eine iterative Methode die begrenzte Konnektivität von Quantum Annealern kompensieren soll. In dieser Arbeit wird keine konkrete Anwendung in der Computer Vision behandelt. Stattdessen werden die entwickelten Optimierungsverfahren auf Max-Cut-Problemen getestet.

Die Doktorarbeit wird kumulativ verfasst, sodass die einzelnen Publikationen separat präsentiert werden. Die Arbeiten sind motiviert durch folgende grundlegende Fragestellungen: Wie groß ist der Nutzen von Quantum Annealing in Computer Vision und wie würden Computer Vision Algorithmen aussehen, welche Quantum Annealing verwenden? Eine andere Sichtweise bietet die folgende Frage: Was würde sich im Forschungsfeld der Computer Vision ändern, wenn es massive Fortschritte im Lösen von quadratischen, unrestringierten, binären Optimierungsproblemen gäbe; unabhängig davon, ob diese mit oder ohne Quantum Computing erzielt werden?

Acknowledgment

First I would like to thank my advisor, Michael Möller. During my whole work as a PhD student he always provided optimal supervision. He always achieved a good balance between providing great research ideas and letting me explore research paths freely. Everything that I know about Computer Vision is due to him. Next, I would like to thank Vladislav Golyanik and Zorah Lähner as the two persons that I collaborated with the most. Both provided invaluable academic support and were also really great people to get to know outside of work hours. I thank all my former colleagues, namely Jonas Geiping, Hartmut Bauermeister, Hannah Dröge, Kanchana Vaishnavi Gandikota, Sarah Wagener, Ulrich Schipper, Kostiantyn Ahapov, Marius Bock, Alexander Auras, Michael Schopf-Küster, Onofre Martorell, Jan Philipp Schneider, Jovita Lukasik and Natacha Kuete Meli. I really enjoyed every chat during lunchtime or coffee breaks. I also want to thank my physics LK teacher Dirk Strohlos, who sadly passed away a few years ago, and my Master thesis supervisors, Otfried Gühne and Gael Sentís. Their kindness and passion for physics and teaching encouraged me to pursue a PhD. I also thank Oliver Wurmbach for being a great friend and for being always fun to hang out with. Finally, I thank my family for their constant and unconditional support. Your encouragement and belief in me made this achievement possible.

„Schöner als die fertige Wahrheit ist das Suchen nach Wahrheit!"

—Oskar Perron, *Über Wahrheit und Irrtum in der Mathematik*, inaugural lecture in Tübingen, 1911

Contents

1 List of Publications **1**

2 Introduction **3**
2.1 Fundamentals of Quantum Mechanics 3
2.1.1 Single Qubits 3
2.1.2 Multi-qubit Systems 4
2.2 Quantum Adiabatic Computing and Quantum Annealing 6
2.2.1 Comparison with Simulated Annealing 11
2.3 Shape Matching Background 12

3 Adiabatic Quantum Graph Matching with Permutation Matrix Constraints **15**
3.1 Contextualization 15
3.2 Introduction 16
3.3 Modern Adiabatic Quantum Computing 18
3.4 Related Work 20
3.5 Minimization of Quadratic Objectives over Permutation Matrices 21
3.5.1 Baseline Quantum Graph Matching 21
3.5.2 The Formulation with a Row-Wise Penalty 22
3.5.3 Inserted Formulation with Eliminated Variables 23
3.6 Experimental Results 24
3.6.1 Evaluation on Random Problem Instances 24
3.6.2 Ablative Study on the Coupling Parameters 26
3.6.3 Experiments with Real Data 27
3.7 Further Observations and Limitations 29
3.8 Conclusions 30
3.9 Supplementary Material for Adiabatic Quantum Graph Matching with Permutation Matrix Constraints 30
3.10 Proofs for Section 3.5 30
3.10.1 Proof of Proposition (3.1) 31
3.10.2 Proofs of Lemma (3.2) and Proposition (3.3) 33
3.11 Experiments on IBM Burlington 35
3.12 Point Cloud Matching Example 36
3.13 Near-Isometric Matching 37
3.14 Embedding to the Chimera Graph 38

3.15 Discussion of Negative Results . . . 38
3.16 Beyond Quantum Computing . . . 39

4 Q-Match: Iterative Shape Matching via Quantum Annealing 43
4.1 Contextualization . . . 43
4.2 Introduction . . . 44
4.3 Related Work . . . 45
4.4 Background . . . 47
4.4.1 Adiabatic Quantum Annealing (AQC)ing . . . 47
4.4.2 Algorithm Design and Programming the D-Wave Quantum Annealer . . . 48
4.4.3 $\boldsymbol{\alpha}$-Expansion . . . 49
4.4.4 Cyclic Permutations . . . 50
4.5 Method . . . 50
4.5.1 Cyclic $\boldsymbol{\alpha}$-Expansion . . . 50
4.5.2 Q-Match . . . 51
4.6 Experiments . . . 53
4.6.1 Comparison to Penalty Based Implementation . . . 54
4.6.2 FAUST . . . 54
4.6.3 Convergence . . . 55
4.6.4 Runtime . . . 55
4.6.5 QAPLIB . . . 56
4.7 Conclusion . . . 57
4.8 Supplementary Material for Q-Match: Iterative Shape Matching via Quantum Annealing . . . 57
4.9 Analysis of the Individual QUBOs . . . 58
4.9.1 Minor Embedding . . . 58
4.10 Derivations and Proofs . . . 61
4.10.1 Derivation of Eq. (4.6) in Sec. 4.5.1 . . . 61
4.10.2 Proof of Lemma 4.1 . . . 62
4.10.3 Simulated Annealing vs. Quantum Annealing for the Subproblems . . . 63
4.11 Failure Case for Individually Optimizing Over the 2-Cycles . . . 65
4.12 Calculation of W_s and $\tilde{W}$ . . . 67
4.12.1 Calculating W_s . . . 67
4.12.2 Calculating $\tilde{W}$ . . . 68
4.13 Exact Solutions on QAPLIB . . . 68
4.14 Non-Optimal Initialization . . . 68

5 CCuantuMM: Cycle-Consistent Quantum-Hybrid Matching of Multiple Shapes 69
5.1 Contextualization . . . 69
5.2 Introduction . . . 70
5.3 Related Work . . . 72
5.4 Background . . . 73
5.4.1 Adiabatic Quantum Computing (AQC) . . . 73
5.4.2 Shape Matching . . . 74
5.4.3 Cyclic $\boldsymbol{\alpha}$-Expansion (CAE) . . . 74

5.5 Our CCuantuMM Method . . . 75
5.5.1 Matching Three Shapes . . . 75
5.5.2 Matching N Shapes . . . 79
5.6 Experimental Evaluation . . . 81
5.6.1 Experiments on Real Quantum Annealer . . . 81
5.6.2 Comparison to Quantum and Classical SoTA . . . 82
5.6.3 Ablation Studies . . . 83
5.6.4 Discussion and Limitations . . . 85
5.7 Conclusion . . . 85
5.8 Supplementary Material . . . 85
5.9 Notations . . . 86
5.10 N-Shape Algorithm . . . 86
5.11 Matching **100** Shapes . . . 86
5.12 Higher-Order Terms . . . 86
5.13 Final QUBO . . . 86
5.14 Evolution During Optimisation . . . 86
5.15 Fixed Anchor vs. Random Triplets . . . 87
5.16 Time Complexity . . . 87
5.17 Implementation Details . . . 90
5.17.1 Initialisation . . . 90
5.17.2 Pre-Computing Trivial Cycles . . . 90
5.18 Minor Embeddings and Other QPU Experiments . . . 91
5.18.1 QPU Processing and Annealing Time . . . 91
5.18.2 Minor Embeddings . . . 91
5.18.3 Minor Embeddings in Practice . . . 92
5.19 Further Results and Ablations . . . 93
5.19.1 Additional Qualitative Results . . . 93
5.19.2 Variation Within a Dataset . . . 93
5.19.3 Influence of Descriptors . . . 93
5.19.4 Noise Perturbation . . . 93
5.20 Related Quantum Computer Vision Works . . . 94

6 QuAnt: Quantum Annealing with Learnt Couplings 105
6.1 Contextualization . . . 105
6.2 Introduction . . . 105
6.3 Related Work . . . 107
6.4 QUBO Learning Approach . . . 108
6.4.1 QUBOs and Quantum Annealing . . . 108
6.4.2 Network Architecture and Losses . . . 108
6.4.3 QUBOs on D-Wave Quantum Annealers . . . 110
6.4.4 Graph Matching . . . 110
6.4.5 Point Set Registration and Rotation Estimation . . . 111
6.5 Experimental Evaluation . . . 111
6.5.1 Results . . . 112
6.5.2 Ablation . . . 113

6.5.3 Evaluation on D-Wave . . . 114
6.5.4 Further Analysis . . . 115
6.6 Discussion . . . 116
6.7 Quantum Computing Background . . . 117
6.7.1 Quantum Annealing in Detail . . . 117
6.7.2 Logical and Physical Qubits . . . 118
6.8 Implementation Details . . . 118
6.9 Graph Matching . . . 119
6.9.1 Problem Description . . . 119
6.9.2 Failure Case . . . 119
6.10 Detailed Comparison with QGM . . . 120
6.11 Solution Quality of SA and QA . . . 121
6.12 3D Rotation Estimation . . . 121
6.12.1 Three Stages (Euler Angles) . . . 121
6.12.2 Variance across Runs . . . 122
6.12.3 Training on Noisy Data . . . 122
6.12.4 Qualitative Comparison on Noisy Test Data . . . 123
6.12.5 Comparison to AQM [111] . . . 123
6.13 2D Point Set Registration . . . 123
6.13.1 Setting Details . . . 123
6.13.2 Variance . . . 124
6.13.3 Noise . . . 124
6.13.4 Qualitative Ablation Results . . . 124
6.13.5 Failure Case . . . 124

7 Quantum Motion Segmentation 127
7.1 Contextualization . . . 127
7.2 Introduction . . . 127
7.3 Background . . . 127
7.3.1 Motion Segmentation . . . 127
7.4 Our Quantum Approach . . . 129
7.4.1 QuMoSeg-v1 . . . 130
7.4.2 QuMoSeg-v2 . . . 132
7.5 Conclusion . . . 133

8 Compensating Connectivity Restrictions in Quantum Annealers via Splitting and Linearization Techniques 135
8.1 Contextualization . . . 135
8.2 Abstract . . . 135
8.3 Introduction . . . 136
8.4 Related Work . . . 137
8.5 Methods . . . 140
8.5.1 Linearization and Damping Term (Splitting) . . . 140
8.5.2 Statement for Ising Problems . . . 142
8.5.3 Fixing Variables, Large Neighborhood Local Search (LNLS) . . . 142

8.5.4 Relation to Projected Gradient Descent . . . 143
8.5.5 Permutating the Adjacency Matrix of the Hardware Graph . . . 144
8.5.6 Strategy for Regularization Parameters . . . 146
8.5.7 Choosing a Part of the Hardware Graph with high Connectivity . . . 146
8.5.8 Simulated Annealing-like Regularization . . . 146
8.6 Results . . . 149
8.6.1 Datasets . . . 149
8.6.2 Regular Spinglas Dataset from Ref. [183] . . . 150
8.6.3 Maximum Cut Problems . . . 150
8.7 Conclusion . . . 154
8.8 Appendix . . . 155
8.8.1 Proof for the Conjecture about Regular Spin Glasses from Ref. [183] . . . 155
8.8.2 Ablation Study . . . 157
8.8.3 MQlib Experiments with k-Opt . . . 158

9 Conclusion and Discussion **161**

Bibliography **163**

Chapter 1

List of Publications

This doctoral thesis is based on following published works:

- Marcel Seelbach Benkner, Vladislav Golyanik, Christian Theobalt, and Michael Moeller, Adiabatic quantum graph matching with permutation matrix constraints, 3DV 2020.
- Marcel Seelbach Benkner, Zorah Lähner, Vladislav Golyanik, Christof Wunderlich, Christian Theobalt, Michael Moeller, Q-Match: Iterative Shape Matching via Quantum Annealing, ICCV 2021.
- Marcel Seelbach Benkner, Maximilian Krahn, Edith Tretschk, Zorah Lähner, Michael Moeller, Vladislav Golyanik, Learning QUBO Forms in Quantum Annealing, ICLR 2023.
- Harshil Bhatia, Edith Tretschk, Zorah Lähner, Marcel Seelbach Benkner, Michael Moeller, Christian Theobalt, Vladislav Golyanik, CCuantuMM: Cycle-Consistent Quantum-Hybrid Matching of Multiple Shapes, CVPR 2023.
- Federica Arrigoni, Willi Menapace, Marcel Seelbach Benkner, Elisa Ricci and Vladislav Golyanik. Quantum Motion Segmentation, ECCV 2022.
- Natacha Kuete Meli, Vladislav Golyanik, Marcel Seelbach Benkner, Michael Moeller, QuCOOP: A Versatile Framework for Solving Composite and Binary-Parametrised Problems on Quantum Annealers, CVPR 2025.

Additionally, the following works are still under review:

- Marcel Seelbach Benkner, Zorah Lähner, Vladislav Golyanik, Martin Kliesch, Michael Moeller, Compensating connectivity restrictions in quantum annealers via splitting and linearization techniques, under review at the Quantum Machine Intelligence journal.
- Natacha Kuete Meli, Shuteng Wang, Marcel Seelbach Benkner, Michele Sasdelli, Tat-Jun Chin, Tolga Birdal, Michael Moeller, Vladislav Golyanik, Quantum-enhanced Computer Vision: Going Beyond Classical Algorithms, unpublished and under review at International Journal of Computer Vision (IJCV).

Chapter 2

Introduction

In the following we will introduce quantum annealing to the reader. This is helpful to understand the reprinted publications and to motivate the research field. In the individual publications one can also find introductions about the physical background. However, due to size constraints, the introduction presented here is much more detailed. This introduction is largely based on the unpublished survey article mentioned in the publication list. For the particular part of the survey that is presented here Marcel Seelbach Benkner was the main contributor.

2.1 Fundamentals of Quantum Mechanics

This chapter introduces the reader to the physical background needed to understand quantum annealing. We will go through some of the fundamentals of quantum mechanics and try to assume only some mathematical knowledge of the reader. In the text we will focus rather on readability than on presenting the most general and precise quantum mechanics introduction. A recommended source to learn more about the topic of quantum computing and quantum information than what is presented here is [197]. Further sources that focus more on quantum annealing and adiabatic quantum computing are [182] and [81]. Although a good understanding of quantum annealing is of course desirable, many of the publications in this thesis can also be comprehended to some degree by treating quantum annealing as black box solver for quadratic, unconstrained, binary optimization problems (QUBOs).

2.1.1 Single Qubits

Let us start with the most fundamental building block in quantum computing, i.e., a *qubit.*

Definition 2.1 (Qubits)**.** The information carrier in a quantum computing system called *qubit* is described by a two-dimensional complex vector of length one, i.e.

$$|\psi\rangle \in \mathbb{C}^2, \ \| \, |\psi\rangle \, \|^2 = 1. \tag{2.1}$$

Note that we adopted the *ket-notation* from physics to write vectors with a bar and a right bracket, and will also adopt the *bra-notation* $\langle\psi| = (|\psi\rangle^*)^T$ to denote the conjugate transpose of

$|\psi\rangle$, such that the multiplication of a bra- and a ket-vector results in their inner product in the $\mathbb{C}^2$ Hilbert space [1]. We can write an arbitrary two dimensional complex vector as a column vector:

$$|\psi\rangle = \begin{bmatrix} a+ib \\ c+id \end{bmatrix}, \tag{2.2}$$

with real coefficients $a, b, c, d \in \mathbb{R}$. The normalization condition is then given by $a^2+b^2+c^2+d^2 = 1$.

Next we discuss the quantum-physical phenomenon of *superposition*. States in superposition do not directly correspond to classical states from experiments. Instead they collapse only after measurement to one of the classical states with a certain probability. This is modeled mathematically in the following way:

Let $|0\rangle \in \mathbb{C}^2$ and $|1\rangle \in \mathbb{C}^2$ be an orthonormal basis of $\mathbb{C}^2$ (whose precise form depends on the setup of the physical system). For example one can have in column vector notation: $|0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, |1\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$.

Definition 2.2 (Measurements)**.** When the state of a qubit $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle \in \mathbb{C}^2$ is *measured* (with respect to the basis $\{|0\rangle, |1\rangle\}$), it results in a value corresponding to

$$\begin{cases} |0\rangle & \text{with probability } |\alpha|^2 = |\langle 0|\psi\rangle|^2, \\ |1\rangle & \text{with probability } |\beta|^2 = |\langle 1|\psi\rangle|^2. \end{cases} \tag{2.3}$$

In other words, although a qubit can be in a superposition of multiple classical states at once, the square of the absolute value of the coefficient in its basis representation $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ determines the probability for measuring the value corresponding to $|0\rangle$ or $|1\rangle$. This is also called "collapse of the wave function". The act of measurement with respect to a basis changes the state into one of the elements of the basis with probabilities defined by the projections of the state onto the basis.

2.1.2 Multi-qubit Systems

When multiple qubits $|\psi_1\rangle, |\psi_2\rangle, \ldots, |\psi_n\rangle$ are considered, then in the simplest case their overall state $|\psi\rangle$ is described by the *tensor or Kronecker product*, i.e., a vector consisting of all combinations of products of the individual coefficients, see [197]:

$$|\psi\rangle = |\psi_1\rangle \otimes |\psi_2\rangle \otimes \ldots |\psi_n\rangle \in \mathbb{C}^{2^n}. \tag{2.4}$$

For two qubits this would look like:

$$|\psi_1\rangle = \begin{bmatrix} \alpha \\ \beta \end{bmatrix}, \ |\psi_2\rangle = \begin{bmatrix} \gamma \\ \delta \end{bmatrix}, \ |\psi_1\rangle \otimes |\psi_2\rangle = \begin{bmatrix} \alpha\gamma \\ \alpha\delta \\ \beta\gamma \\ \beta\delta \end{bmatrix}. \tag{2.5}$$

[1] Note that in quantum mechanics states can also have higher dimension than qubits. For non discrete quantities like momentum or position these states are not finite dimensional anymore but functions from the Hilbert space $\mathcal{L}^2(\mathbb{R}^3)$. In this case the bra vectors can be understood as functionals from $\mathcal{L}^2$. Even the formulation on $\mathcal{L}^2$ still has shortcomings and a full mathematical treatment would have to resort to rigged Hilbert spaces to describe distributions and deal with unbounded operators [16].

A useful shorthand for the Kronecker product is to rewrite

$$|\psi_1\rangle \otimes |\psi_2\rangle \otimes \dots |\psi_n\rangle = |\psi_1\psi_2 \dots \psi_n\rangle\,, \tag{2.6}$$

hence, for example, the third equation in (2.5) becomes

$$|\psi_1\psi_2\rangle = \begin{bmatrix} \alpha\gamma \\ \alpha\delta \\ \beta\gamma \\ \beta\delta \end{bmatrix}. \tag{2.7}$$

A surprising feature of quantum mechanics is that the state of the multi-qubit system is not limited to this low dimensional subset of only those 2^n dimensional vectors that can be decomposed into a tensor product (2.4). In fact every normalized vector in $\mathbb{C}^{2^n}$ is a physically meaningful state vector describing the whole system and the concept of *entanglement* says that systems of multiple qubits are **not** limited to so-called *separable* states that decompose as (2.4).

The state

$$\frac{|01\rangle + |10\rangle}{\sqrt{2}} = \begin{bmatrix} 0 \\ 1/\sqrt{2} \\ 1/\sqrt{2} \\ 0 \end{bmatrix} \tag{2.8}$$

for example cannot be decomposed as

$$|\psi_0\psi_1\rangle = \begin{bmatrix} \alpha\gamma \\ \alpha\delta \\ \beta\gamma \\ \beta\delta \end{bmatrix} \tag{2.9}$$

since there are no $\alpha, \beta, \gamma, \delta$ that can simultaneously satisfy the equations $\alpha\gamma = 0,\ \alpha\delta = 1/\sqrt{2},\ \beta\gamma = 1/\sqrt{2},\ \beta\delta = 0$. This is one of the famous Einstein-Podolsky-Rosen states [197]. The following definition of entangled states is essential in quantum mechanics because it models certain non-classical correlations between subsystems.

Definition 2.3 (Entanglement)**.** A system of n many qubits can evolve into *any* state in $\mathbb{C}^{2^n}$. States that can be represented via (2.4) are called *separable*, while states that do not admit such a representation are called *entangled*.

One way to interpret this is, is that for entangled states the whole multi-qubit system is not fully determined by just the states of the individual qubits. Notice also that if one writes down the coefficients for a bipartite system as matrix, then the separable states are exactly those where the coefficient matrix has the rank 1.

Furthermore, it is possible to manipulate the state of an n-qubit quantum physical system experimentally, e.g., prepare an initial state $|\psi(0)\rangle$ and manipulate it with external influences. These influences are incorporated in a description of the energy of the system, which can be obtained via calculating the eigenvalues of a certain, complex matrix $H(t) \in \mathbb{C}^{2^n \times 2^n}$. The matrix

depends on the time t and determines the system's energies for example by describing if the n-many qubits experience a certain coupling. The way the state vector evolves in time depends on this description of possible energies of the system as can be seen in the famous Schrödinger Equation.

Definition 2.4 (Schrödinger Equation)**.** The (time) evolution of a quantum state $|\psi(t)\rangle$ is described by the *Schrödinger Equation*

$$i\hbar\frac{d}{dt}|\psi(t)\rangle = H(t)|\psi(t)\rangle, \tag{2.10}$$

where the *Hamiltonian* $H(t) \in \mathbb{C}^{2^n \times 2^n}$ is a Hermitian matrix determined by the experimental setup, i is the imaginary number, and $\hbar$ is the reduced Planck constant.

A closer look into the solutions of this differential equation reveals that they follow some unitary time evolution [197]. This is in particular consistent with what we learned that quantum states have norm 1, since the length of the components is related to probabilities.

2.2 Quantum Adiabatic Computing and Quantum Annealing

In contrast to constructing a unitary time evolution with a set of gates as it is done in gate-based quantum computing one can also imagine systems where the unitary time evolution depends rather continuously on the time. One such model of quantum computing called adiabatic quantum computing will be described in this section. We also show the proof of the adiabatic theorem, which supports quantum annealing as a heuristic for solving QUBO problems. A proof like this can also be found in [81] or in references cited in [123], while the main text from [123] also provides further useful context.

To describe the method of quantum annealing one has to specify the time dependent Hamilton operators from the Schrödinger equation and the initial state. We start by stating the fact that for certain fixed initial Hamiltonians $H(0) = H_I$ it is physically possible to create an initial quantum state $|\psi(0)\rangle$ that is an eigenvector to the smallest eigenvalue of the operator. Moreover, it is possible (with certain system dependent limitations) to create n^2 many interactions $J_{i,j} \in \mathbb{R}$ between the individual qubits along with n biases $b_i \in \mathbb{R}$ acting on each qubit that amount to a Hamiltonian of the form

$$H_P = \sum_{i,j} J_{i,j}\sigma_i^z\sigma_j^z + \sum_i b_i\sigma_i^z, \tag{2.11}$$

$$\sigma_i^z = \underbrace{I \otimes I \ldots \otimes I}_{i-1\text{-many times}} \otimes\sigma_z \otimes \underbrace{I \otimes \ldots \otimes I}_{n-i\text{-many times}}, \tag{2.12}$$

$$\text{for } \sigma_z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}, \quad I = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}. \tag{2.13}$$

One can show that these $n^2 + n$ many terms lead to a Hamiltonian H_P that is a diagonal $2^n \times 2^n$ matrix whose diagonal entries are the enumeration of all costs obtained via $s^T Js + s^T b$ for $s \in \{-1, 1\}^n$. Thanks to this enumeration, the eigenvector $|\psi\rangle$ to the smallest eigenvalue of H_P is a unit vector whose only entry equal to one can be identified with a particular $s \in \{-1, 1\}^n$

that is the solution to

$$\min_{s\in\{-1,1\}^n} s^T Js + s^T b. \tag{2.14}$$

The central idea of adiabatic quantum computing then merely follows from the free evolution of the state vector $|\psi(t)\rangle$ (Assumption 2.3) under the Schrödinger equation (Assumption 2.4) with an experimentally constructed Hamiltonian of a form like

$$H(t) = (1 - f(t))H_I + f(t)H_P \tag{2.15}$$

for a function $f\colon [0,T] \to [0,1]$ that slowly transitions between the two Hamiltonians in time T, e.g., $f(t) = t/T$:

Theorem 2.5 (Quantum Adiabatic Theorem)**.** *The minimum time necessary for an adiabatic transition can be thought as the runtime of the quantum algorithm. If a system evolves under the equation* (2.10) *starting from an eigenvector $|\psi(0)\rangle$ to H_I, then a sufficiently slow transition in* (2.15) *ensures that the systems $|\psi(t)\rangle$ remains an eigenvector of $H(t)$ during the entire evolution until $t = T$, if its corresponding eigenvalue retains a sufficiently large gap to all other eigenvalues of $H(t)$ during the entire evolution.*

In summary one first has to prepare the system in an eigenstate of H_I corresponding to the smallest eigenvalue. After that one evolves H_I to the problem Hamiltonian (2.11) sufficiently slow via (2.15). In the end measuring the state after such an evolution should yield minimizers of (2.14) (or at least be an excellent metaheuristic for computing this).

Let us prove that this is indeed merely a mathematical consequence of our assumptions.

Proof. Note that all $H(t)$ are Hermitian matrices, such that for all t there exists an orthonormal basis $|k(t)\rangle$ of eigenvectors to eigenvalues $\lambda_k(t)$, i.e.

$$H(t)\,|k(t)\rangle = \lambda_k(t)\,|k(t)\rangle\,. \tag{2.16}$$

Thus, at each time we can represent a quantum state $|\psi(t)\rangle$ that is evolving under (2.10) as

$$|\psi(t)\rangle = \sum_k c_k(t)\,|k(t)\rangle \tag{2.17}$$

for suitable coefficients $c_k(t)$. Subsequently, we insert this expansion into the Schrödinger equation and show that it results in a component-wise equation in the coefficients of the (time-dependent) eigenbasis,

$$i\hbar(c_k'(t) + \langle k(t), k'(t)\rangle)c_k(t) \approx \lambda_k(t)c_k(t), \tag{2.18}$$

up to a sum over terms involving $\langle l(t), k'(t)\rangle$, $l \neq k$. Note that if (2.18) held exactly, then

$$c_k(t) = c_k(0)\exp\left(-\frac{i}{\hbar}\int_0^t \lambda_k(s) - i\hbar\langle k(s), k'(s)\rangle\, ds\right) \tag{2.19}$$

would yield the claim. Another important point is that the exponent is purely imaginary. For the second term in the integral this can be verified by differentiating the norm of $k(s)$ which should

be zero and analyzing the result. To obtain a bound on $\langle l(t), k'(t)\rangle$ one can differentiate (2.16) and take the product with $\langle l(t)|$ to obtain

$$\langle l(t), k'(t)\rangle = \frac{\langle l(t), H'(t)k(t)\rangle}{\lambda_l(t) - \lambda_k(t)}, \tag{2.20}$$

which is small if the change of the Hamiltonian H' is small in comparison to all gaps of eigenvalues $\lambda_l(t) - \lambda_k(t)$. In particular, crossing eigenvalues, i.e., $\lambda_l(t) = \lambda_k(t)$ for some time t, makes the claim invalid. □

As we can see from the above proof, an important quantity for the idea of quantum annealing is the smallest difference of the second smallest to the smallest eigenvalue of $H(t)$ during the evolution (2.15), which is also called the *spectral gap*.
A common choice for the initial operator H_I is

$$H_I = -\sum_i \kappa\sigma_i^x \tag{2.21}$$

$$\sigma_i^x = \underbrace{I \otimes I \ldots I}_{i-1\text{-many times}} \otimes\sigma_x \otimes \underbrace{I \otimes I \ldots \otimes I}_{n-i\text{-many times}}, \tag{2.22}$$

$$\text{for } \sigma_x = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad I = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \tag{2.23}$$

with a constant $\kappa \in \mathbb{R}$. Under rather general conditions one can then show that the spectral gap (and thus the denominator in (2.20)) does not become zero, if the solution to the original problem (2.14) is unique.

Theorem 2.6 (No crossing of eigenvalues)**.** *For H_P given by Eq.* (2.11) *and H_I by Eq.* (2.21) *the lowest and the second lowest eigenvalue of $H(t)$ from Eq.* (2.15) *do not cross during the time evolution, assuming that the ground state of H_P is not degenerate.*

Proof. The basic idea is to show that the smallest eigenvalue of $H(t)$ is simple using the Perron-Frobenius Theorem. Before we come to the actual proof, let us recall a particular part of the Perron-Frobenius Theorem.

Theorem 2.7 (Perron-Frobenius)**.** *Let $H \in \mathbb{R}^{m\times m}$ be non-negative, i.e., all $H_{i,j} \geq 0$, and irreducible. Then the largest eigenvalue in magnitude is positive and simple.*

Our proof strategy is to show that $-H(t)$ in the evolution of the Hamiltonians is non-negative and irreducible, such that the simplicity of the largest eigenvalue ensures a non-zero spectral gap, i.e., a non-zero difference between the smallest and second smallest eigenvalues of $H(t)$.

Recall that a matrix $H \in \mathbb{R}^{m\times m}$ is irreducible if and only if the directed graph we obtain by looking at vertices i with $i \in \{1, \ldots, m\}$ and connecting vertex i with vertex j if $H_{i,j} \neq 0$, is *strongly connected*. The latter means that there is a path from any vertex to any other vertex. We first prove the following lemma.

Lemma 2.8. *The matrix H_I defined as*

$$H_I = -\sum_i \kappa\sigma_i^x \tag{2.24}$$

$$\sigma_i^x = \underbrace{I \otimes I \ldots I}_{(i-1)\text{-many times}} \otimes \sigma_x \otimes \underbrace{I \otimes I \ldots \otimes I}_{(n-i)\text{-many times}}, \tag{2.25}$$

$$\text{for } \sigma_x = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \quad I = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{2.26}$$

is irreducible.

Proof. For any $a \in \{-1, 1\}$, define

$$e(a) = \begin{cases} \begin{bmatrix} 1 \\ 0 \end{bmatrix} & \text{if } a = 1, \\ \begin{bmatrix} 0 \\ 1 \end{bmatrix} & \text{if } a = -1. \end{cases}$$

and for any $s \in \{-1, 1\}^n$ consider $e_s := e(s_1) \otimes e(s_2) \otimes \ldots \otimes e(s_n)$. We find that

$$\begin{aligned} \sigma_i^x e_s &= e(s_1) \otimes \ldots \otimes e(s_{i-1}) \otimes \sigma_x e(s_i) \otimes e(s_{i+1}) \otimes \ldots \otimes e(s_n) \\ &= e(s_1) \otimes \ldots \otimes e(s_{i-1}) \otimes e(-s_i) \otimes e(s_{i+1}) \otimes \ldots \otimes e(s_n). \end{aligned}$$

This means, interpreting the 2^n many indices of H_I as edges associated with every $s \in \{-1, 1\}^n$, the non-zero entries of H_I connect every vertex s to any other vertex t that arises from s by flipping one sign. Since every $s^{(1)}$ can be generated from every other $s^{(2)}$ by a sequence of sign flips, the graph induced by H_I is strongly connected which yields the assertion. Note also that the induced graph of H_I is a hypercube graph. □

We can now prove the intended theorem 2.6:
Proof. First, note that the diagonal entries of H_I are zero. Second, note that H_P is a diagonal matrix. Third, note that for any $c \in \mathbb{R}$

$$\begin{aligned} &\operatorname*{arg\,min}_{s \in \{-1,1\}^n} s^T J s + s^T b && (2.27) \\ &= \operatorname*{arg\,min}_{s \in \{-1,1\}^n} s^T J s + s^T b - c \cdot n && (2.28) \\ &= \operatorname*{arg\,min}_{s \in \{-1,1\}^n} s^T J s + s^T b - c(s^T I s) && (2.29) \\ &= \operatorname*{arg\,min}_{s \in \{-1,1\}^n} s^T (J - cI) s + s^T b, && (2.30) \end{aligned}$$

which means that one can always ensure all entries of H_P to be negative. Thus, for any continuous, monotonically increasing $f\colon [0, T] \to [0, 1]$ with $f(0) = 0$ and $f(T) = 1$ we have that

$$H(t) = (1 - f(t))H_I + f(t)H_P$$

is non-positive and irreducible for $t < T$ due to the Perron-Frobenius Theorem (and the previous Lemma), which means that the smallest eigenvalue is simple. Finally, the assumption of the ground state being non-degenerate, yields the same for $t = T$. Thus, the minimal spectral gap

in $[0, T]$ (which is attained as a continuous function over a compact interval) is greater than zero. □

Note that the constant κ in the definition of H_I allows to scale all eigenvalues and has an effect on the speed $H'(t)$ of the transition in (2.15).

While a zero spectral gap can be avoided for all problems with unique solutions, the question how the total annealing time T necessary for an *adiabatic transition*, i.e., one where the system remains in its ground state, scales with the dimension of the overall problem, is crucial for the complexity of the presented quantum annealing algorithm. In particular, if the spectral gap becomes exponentially small, one will need an exponential increase in the annealing time T (indicating a very difficult problem even for a quantum computer). Since quadratic, unconstrained, binary optimization problems (QUBOs) are NP-hard in general, it is believed that even quantum computers will require an exponentially increasing runtime for solving them in general. For a linear transition between H_I and H_P in (2.15) it was proven that certain instances indeed require an exponentially increasing runtime [259]. On the topic of how to change the annealing path and the Hamiltonians to avoid exponentially small gaps, if possible, a lot of research has been done [11]. Another interesting research area is how the success probability depends on the amount of entanglement the state $|\psi(t)\rangle$ has for $t \in (0, T)$. In [28] it was shown that quantum systems that give a speedup via adiabatic quantum computing can have a low amount of multipartite entanglement. Therefore the relation between those quantities is complex and one cannot claim that entanglement fully explains the performance of AQC algorithms. Another direction of research is to deliberately deviate from a sufficiently slow transition of Hamiltonians in order to yield Baysian estimates rather than optimal solutions, see e.g. [181]. As an exhaustive discussion on adiabatic quantum computing itself goes beyond the scope of this work, we refer the interested reader to the overview [123] and rather focus on some practical aspects relevant to the publications.

The Canadian company D-Wave currently builds the largest quantum annealing systems. They provide remote access to their quantum annealers via a cloud service [72]. For this thesis experiments were performed on quantum processors of two generations. The quantum devices are called 2000Q and Advantage. 2000Q contains ≈2000 qubits, with each qubit within a unit cell (of eight qubits) being able to couple with four other qubits. The Advantage generation provides ≈5500 qubits, with each qubit within a unit cell (of 24 qubits) connected to fifteen other qubits [84]. The exact form of the hardware graph can be seen in Fig. 2.1.

The connectivity patterns of 2000Q and Advantage restrict how problems formulated in terms of logical qubits (*logical* means qubit variables that are abstracted from physical qubit realisations) are mapped to hardware. *Minor embedding* algorithms (such as Cai *et al.* [52]) can calculate a mapping from the logical problem graph to the physical hardware graph.

Most current quantum annealing experiments can only be used to solve QUBO problems, but it should also be stressed that by considering different Hamiltonians other optimization problems may also work.

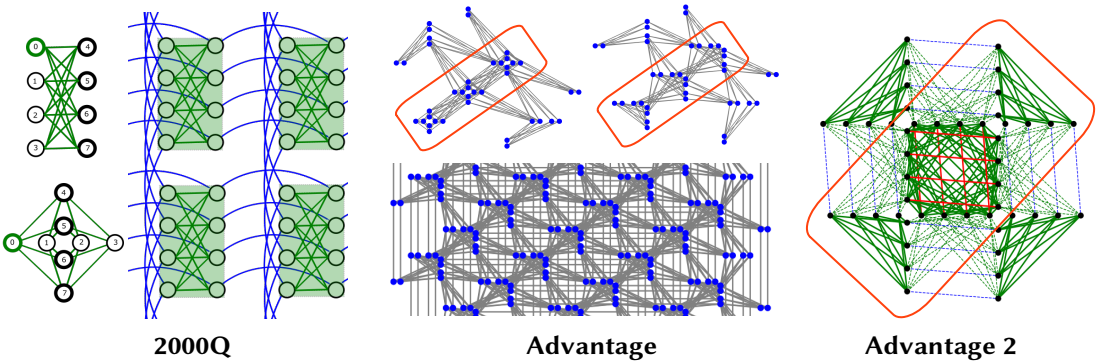


Figure 2.1: Hardware graphs of the different D-Wave architectures. The figure is taken from the survey article in preparation.

2.2.1 Comparison with Simulated Annealing

A comparison between quantum annealing and classical optimisation algorithms is also insightful. One algorithm similar to quantum annealing is simulated annealing. The simulated annealing algorithm works iteratively. In the beginning of a step a random neighbor from the current iterate is chosen. If the energy of the randomly chosen neighbour is better than the current iterate, it gets accepted and is the new iterate. If the energy gets worse, the randomly chosen neighbour is only accepted with probability

$$P_{\text{accept}} = \min\{\exp\Big((E(x^{(i-1)}) - E(x^{(i\text{-candidate})}))/T_i\Big)\} \tag{2.31}$$

The sequence T_i is called temperature schedule and is monotonically decreasing with every iteration step i. As a results it becomes progressively more unlikely that worse solutions are accepted. An example for a temperature schedule is $T_i = \frac{c}{\log(i+1)}$ with a problem dependent constant c [110]. For this temperature schedule convergence results to the global optimum in a similar manner like Theorem 2.5 exist [121]. However in practical implementations one often uses schedules that converge faster [173].

At some point, the algorithm will typically end up in a local optimum that it cannot escape from anymore. A pseudo code for the algorithm can be seen in Algorithm 1.

Algorithm 1 Simulated Annealing algorithm

Input: Energy function E to be optimized, Temperature schedule T_i, neighbourhood structure $\mathcal{N}(\cdot)$, *maxiter*
Output: Solution $x^{(maxiter)}$

1: Choose $x^{(0)}$
2: **for** i=1 to *maxiter* **do**
3: Draw random neighbour: $x^{(i\text{-candidate})} \in \mathcal{N}(x^{(i-1)})$
4: $x^{(i)} = x^{(i\text{-candidate})}$ with probability
5: $P_{\text{accept}} = \min\{1, \exp\big((E(x^{(i-1)}) - E(x^{(i\text{-candidate})}))/T_i\big)\}$
6: **end for**
7: Return $x^{(maxiter)}$

An early research question for adiabatic quantum computation was for which energy potentials it is beneficial compared to simulated annealing. In [100] an energy landscape with a spike is presented, where simulated annealing takes exponential time to get to the global optimum, while adiabatic quantum computation finds the global optimum in polynomial time.

Another question one may ask is how quantum annealing or simulated annealing compare to classical algorithms that are inspired by quantum annealing. This question is explored for example in [67]. Here an algorithm is investigated where low energy states are estimated using classical Markov chain Monte Carlo methods. This algorithm has an exponential advantage to simulated annealing on a problem class with a high spike where quantum annealing also exponentially outperforms simulated annealing. Thereby it is ruled out for these problems that advantages to simulated annealing that were explained with quantum tunneling can not be replicated with classical algorithms. It is important to note, that although the authors name this Monte Carlo algorithm "simulated quantum annealing", the quantum annealing process is not really simulated [13]. Instead quantum Monte Carlo methods are rather used to obtain the thermal state of the Hamiltonian.

This can be done particularly well for Hamiltonians that are stoquastic, i.e. the off-diagonal entries are real and non-positive (in the standard basis). The underlying Hamiltonians for the D-Wave machines are also stoquastic. Nevertheless there exist already experiments were a scaling advantage towards simulated quantum annealing was reported [147].

2.3 Shape Matching Background

Multiple of the presented publications in this thesis are concerned with the shape matching problem. The task of shape matching gained a lot of focus in the computer vision and graphics community with increased attention towards 3D data. One key development was the technological innovation to obtain accurate 3D scans of persons [41]. But even before there was active research about computing correspondences between two 3 dimensional shapes [261]. A key property for the mathematical formulation of the shape matching problem is if they are approximately isometric. This means that there exists a mapping between the shapes so that the geodesic distance, i.e. the shortest distance on the surface between two points, is preserved. An example could be pose changes of a person. No matter in which pose the person is recorded, the shortest distance on the surface between the nose tip and the tip of the thumb will to a very good approximation stay the same. Let us assume that the shapes are described by sets of 3D points $\{x_i | i \in \{1, ..., N\}\}$ and $\{y_i | i \in \{1, ..., N\}\}$. Further let $\hat{\pi} : \{1, ..., N\} \to \{1, ..., N\}$ be the correct permutation for mapping the point clouds to each other. That this mapping is an isometry means for the geodesic distances on the shapes $\mathcal{X}$, $\mathcal{Y}$ denoted as $d_{\mathcal{X}}, d_{\mathcal{Y}} : \{1, ..., N\}^2 \to \mathbb{R}$ that

$$\forall i, j \in \{1, ..., N\} \quad d_{\mathcal{X}}(i, j) = d_{\mathcal{Y}}(\hat{\pi}(i), \hat{\pi}(j)) \tag{2.32}$$

will hold. Motivated by the assumption that the true correspondence would yield an isometric matching one can investigate the following optimization problem

$$\underset{\text{Permutations } \pi}{\arg\min} \sum_i \sum_j |d_{\mathcal{X}}(i, j) - d_{\mathcal{Y}}(\pi(i), \pi(j))|. \tag{2.33}$$

Note however that the solution to this optimization problem might not be unique. For example in cases with some symmetry there can also exist multiple isometries.

If we write π as a permutation matrix X, the equation (2.33) can be formulated as a quadratic assignment problem. We can write the optimization problem as

$$\min_{X \in \mathbb{P}_n} \; E(X) := \mathbf{x}^{\mathrm{T}} W \mathbf{x}, \tag{2.34}$$

where $\mathbf{x} = \mathrm{vec}(X)$ is the n^2-dimensional vector corresponding to a matrix in the set of $n \times n$-dimensional permutation matrices $\mathbb{P}_n \subset \{0,1\}^{n \times n}$ and $W \in \mathbb{R}^{n^2 \times n^2}$ is defined as

$$W_{i \cdot n + k, j \cdot n + l} := |d_{\mathcal{X}}(i,j) - d_{\mathcal{Y}}(k,l)| \tag{2.35}$$

where i, j are vertices on $\mathcal{X}$, k, l are vertices on $\mathcal{Y}$. The vectorization of the matrix X can work in the way that the matrix entry (i,j) is written as the $(i \cdot n + j)$-th entry of the vector x.

If one has a graph describing the shape, the geodesic distance can be computed by using Dijkstra's algorithm. In this algorithm one wants to find the shortest path between two nodes by multiple times updating the distance to the starting node. Computing all the geodesic distances for a dense mesh with many points can become very time consuming nevertheless.

Another way to investigate correspondences of shapes is via heat kernels. In our works here they are in particular used to generate features for an initial matching. The idea is to solve the heat diffusion equation on the meshes. As an initial heat distribution one typically uses a delta distribution, so that the heat is concentrated in one point. By solving the differential equation one can predict how heat propagates through the shape. An advantage of an analysis of the shape in this kind of way is that it is very tolerant of noise in the vertex positions [265].

As already mentioned, optimization problems of the form (2.34) are called quadratic assignment problems (QAP). They arise in a multitude of applications for example in optimizing keyboard layouts or designing facilities like hospitals to reduce the lengths of common paths [51, 199]. Since these problems are $\mathcal{NP}$-hard, research in powerful heuristics is very important. In particular it is not possible to have a constant factor approximation algorithm that is polynomial in time if $\mathcal{P} \neq \mathcal{NP}$ [112, 231].

Describing the correspondence between shapes as a permutation between vertices poses also some fundamental issues or challenges. It is often difficult to match vertices that are close to each other and it can happen that there are problems due to the discretization of the mesh. If one only looks at geodesic distances, symmtries can also be problematic and could lead to a swap between the right and the left side of a human shape. Further hard tasks in shape matching occur if there are large deformations between the shapes. Finally it can even happen that the topology changes if for example the hand touches the body in one of the two poses [97].

Chapter 3

Adiabatic Quantum Graph Matching with Permutation Matrix Constraints

3.1 Contextualization

In this chapter, the 3DV publication "Adiabatic Quantum Graph Matching with Permutation Matrix Constraints" is reprinted. Graph matching is a very important problem in Computer Vision, since it usually occurs if one wants to match key points from two images. The task of shape matching can also be formulated as a graph matching problem. Several of the other publications presented in this thesis can be viewed as some extension to the idea of this first publication. Therefore, this work is central to the thesis. While the concept of finding analytical bounds for the penalty parameter is, from a theoretical perspective, very interesting, it has some drawbacks in experiments. In this work we also noticed that penalty factors that are too large could be problematic because experimental errors effectively limit the precision of the couplings.

Michael Moeller had the idea for this project. Marcel Seelbach Benkner carried out the experiments and derived the penalty parameter bounds. Vladislav Golyanik contributed along the way and had the idea to perform experiments with the quantum annealer. He also provided valuable supervision for finalizing the work before the submission. Christian Theobalt provided helpful discussion.

Abstract

Matching problems on 3D shapes and images are challenging as they are frequently formulated as combinatorial quadratic assignment problems (QAPs) with permutation matrix constraints, which are $\mathcal{NP}$-hard. In this work, we address such problems with emerging quantum computing technology and propose several reformulations of QAPs as unconstrained problems suitable for efficient execution on quantum hardware. We investigate several ways to inject permutation matrix constraints in a quadratic unconstrained binary optimization problem which can be mapped to quantum hardware. We focus on obtaining a sufficient *spectral gap*, which further increases the probability to measure optimal solutions and valid permutation matrices in a single run. We perform our experiments on the quantum computer D-Wave 2000Q (2^{11} qubits,

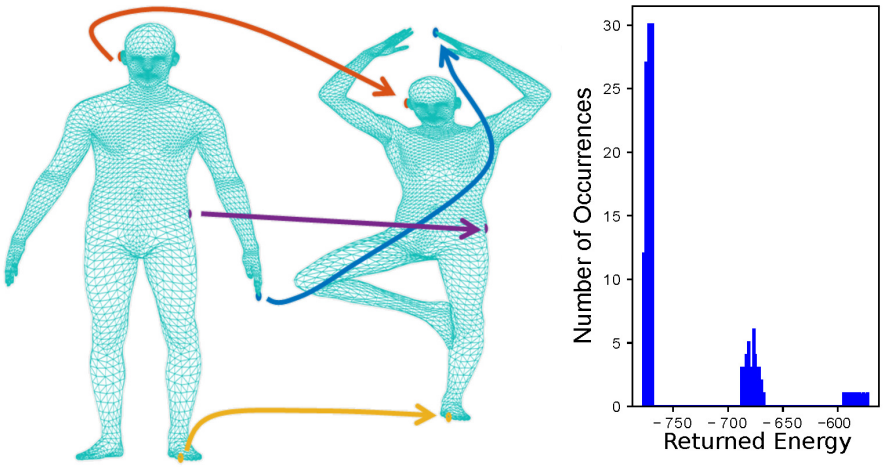


Figure 3.1: Example of 3D shape matching on quantum annealer D-Wave 2000Q. **(Left:)** 3D Shapes from the FAUST dataset [41] and the matches extracted with our QGM. **(Right:)** Histogram of solutions on D-Wave 2000Q. The peaks in the solution distribution arise from the proposed regularization which enhances the chance to measure a valid permutation matrix. The left (highest) bin corresponds to valid permutations. The solutions with the lowest energy, *i.e.,* the left-most measured values in the histogram, correspond to the recovered permutation in the picture.

adiabatic). Despite the observed discrepancy between simulated adiabatic quantum computing and execution on real quantum hardware, our reformulation of permutation matrix constraints increases the robustness of the numerical computations over other penalty approaches in our experiments. The proposed algorithm has the potential to scale to higher dimensions on future quantum computing architectures, which opens up multiple new directions for solving matching problems in 3D computer vision and graphics[1].

3.2 Introduction

The question how quantum computing can be beneficial in solving problems in computer vision is still relatively unexplored. A quantum computer is a computing machine which takes advantage of quantum effects, *i.e.,* quantum superposition, entanglement, tunnelling and contextuality [197, 129]. Since late 1980s, the quantum computing paradigm attracts more and more attention of computer scientists. Multiple quantum methods were subsequently shown to improve the computational complexity [35, 243, 241, 116, 260] compared to their classical algorithmic counterparts.

Only recently the practicability of all these methods has been confirmed [21]. Thus, it is only a few years since quantum computing technology became mature enough and the first functional quantum computers suitable for real-world problems became available for research purposes for a broader community [1, 70, 191, 64, 111].

[1] visit `http://gvv.mpi-inf.mpg.de/projects/QGM/` (the source code is available)

Our motivation in this paper is to investigate the applicability of modern and upcoming adiabatic quantum computers (AQC-ers) for computer vision and graphics tasks. Therefore, we choose the fundamental problem of visual computing *i.e.,* combinatorial graph matching (CGM). This problem allows for a general formulation and, on the other hand, is challenging enough to map on a quantum computer, for the reasons to become clear in the following.

Many matching problems on 2D, 3D and higher-dimensional structures such as point set and mesh alignment [140, 175] require optimizing a quadratic cost function over the set of permutation matrices, *i.e.,* solving a quadratic assignment problem (QAP)

$$\min_{X \in \mathbb{P}_n} \quad f(\mathbf{x}) := \mathbf{x}^{\mathrm{T}} W \mathbf{x} + \mathbf{c}^T \mathbf{x}, \tag{3.1}$$

where $\mathbf{x} = \mathrm{vec}(X)$ is the n^2-dimensional vector corresponding to the matrix $X \in \mathbb{R}^{n \times n}$, which is constrained to lie in the set of permutation matrices $\mathbb{P}_n$. While there exist several well-working heuristics and approximate solvers for (3.1) (see Sec. 5.3 for a summary), there are problem instances on which every such approach would fail to provide optimal results within reasonable computational time as to be expected due to the $\mathcal{NP}$-hard nature of QAPs.

Motivation and Contributions. With adiabatic quantum computing (AQC-ing) moving from theoretical considerations to actual implementations [208, 57], it becomes increasingly attractive for tackling $\mathcal{NP}$-hard problems in computer vision. Constrained problems like (3.1) can unfortunately not be solved with AQC-ers directly, and have to be converted to a quadratic *unconstrained* binary optimization problem (QUBO). We observe that a straightforward reformulation of the constrained problem as a QUBO via a quadratic penalty increases the difficulty of the resulting problem significantly. We conjecture that this difficulty is largely related to the problem's property called the *spectral gap*, *i.e.,* the difference between the lowest and second-lowest energy state, and tailor our approaches at maximizing the latter while still provably solving the original constrained problem. To summarize, our **contributions** are:

- We show how quadratic assignment problems (3.1) with permutation matrix constraints can be mapped to QUBOs and efficiently solved with quantum annealing for small problem instances (Sec. 3.5), see Fig. 3.1 for an example of 3D shape registration. We call our method *Quantum Graph Matching* (QGM). It opens up new opportunities for multiple problem types in 3D computer vision, with a potentially high impact for the future generation of quantum annealers. We impose the permutation matrix constraints not in a post-processing step (*e.g.,* by projecting a relaxed solution to the space of permutation matrices), but directly through problem weights between the qubits, which leads to high probabilities of measuring solutions corresponding to valid permutation matrices.
- The probability to obtain globally-optimal solutions depends on 1) how the problem is mapped to a QUBO and 2) what is the spectral gap of the mapping. We thus propose different approaches to map our formulation to a QUBO and perform spectral gap analysis (Sec. 3.6). Our new way to impose soft permutation matrix constraints is controllable by a parameter λ per each row of the permutation matrix.
- Numerical verification in simulated experiments as well as on a real AQC-er, which shows that the proposed methods effectively increase the success rate of solving combinatorial optimization problems with permutation matrix constraints (Sec. 3.6). Despite several works related to machine learning and image classification [193, 8, 206, 8], where experiments on quantum

hardware were performed, and some theoretical works in quantum computing for computer vision [194, 196, 267, 61, 143], we are not aware of previous AQC-ing experiments for image and shape matching.

In the rest of the paper, Sec. 3.3 discusses the foundations of modern adiabatic quantum computing. Sec. 5.3 reviews related works. Three variants of our QGM based on minimization of quadratic objectives over permutation matrices with AQC-ing are introduced in Sec. 3.5. Sec. 3.6 elaborates on numerical results with simulated and real data, followed by a Discussion 6.6 and Conclusion 3.8.

3.3 Modern Adiabatic Quantum Computing

We next review the basics of AQC-ing. We assume that the reader is familiar with general concepts of linear and tensor algebra, calculus as well as operator theory. No preliminary knowledge of quantum physics is assumed from the reader.

State Space of Quantum Systems. While the state of a quantum-mechanical system is generally described by a normalized vector in a Hilbert-space $\mathcal{H}$, a simple example for a quantum mechanical property is the spin of electrons, in which case the Hilbert-space $\mathcal{H}$ is the two-dimensional complex Euclidean space, $\mathcal{H} = \mathbb{C}^2$. The corresponding quantum state, *i.e.,* an element of the Hilbert space with norm 1, commonly denoted as $|\psi\rangle \in \mathbb{C}^2$, is called a *qubit*. For vectors $|0\rangle, |1\rangle$[2] that form an orthonormal basis of $\mathbb{C}^2$, any such quantum state can be represented as a linear combination of the basis vectors, *i.e.,*

$$|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle\,,\ \ \alpha,\beta \in \mathbb{C},\ \ |\alpha|^2 + |\beta|^2 = 1. \tag{3.2}$$

Quantum mechanics dictates that measuring a qubit in the basis $|0\rangle, |1\rangle$ yields the states $|0\rangle$ and $|1\rangle$ with the probability $|\alpha|^2$ and $|\beta|^2$, respectively. The measurement itself influences the state such that it collapses to either $|\psi\rangle = |0\rangle$ or $|\psi\rangle = |1\rangle$. The state with $|\alpha|^2 = |\beta|^2 = 0.5$ is denoted by $|+\rangle$. The state $|\psi\rangle$ for $\alpha, \beta \neq 0$ is called a *superposition*.

Composite Quantum Systems. Currently, a common approach to realize a quantum computer is to build experiments with multiple qubits, in which case the Hilbert space for representing the overall system is expressed as the tensor product of the individual Hilbert spaces. If one now thinks of multiple spin-$\frac{1}{2}$ particles or n qubits, we obtain states $|\psi\rangle \in \mathcal{H}\left(\mathbb{C}^2\right)^{\otimes n} := \mathbb{C}^2 \otimes ... \otimes \mathbb{C}^2$. Note that if the spins of the electrons were completely independent from each other, we would only need $2n$ many parameters to describe a state. The reason we need the 2^n parameters for the state space is that for so called *entangled states* the spins of the individual systems cannot be treated separately.

Time Evolution of Quantum Systems. The time evolution of a quantum system, *i.e.,* description how a quantum state $|\psi\rangle$ changes over time under an external influence, is given by the Schrödinger equation:

$$i\hbar\frac{\partial}{\partial t}\,|\psi(t)\rangle = H(t)\,|\psi(t)\rangle\,, \tag{3.3}$$

[2] we use Dirac notation; $|0\rangle$ and $|1\rangle$ represent an orthonormal basis

where i is the imaginary unit, $\hbar$ is the reduced Planck constant, t is the time and $H(t)$ is the so-called *Hamiltonian*, which is a self-adjoint, linear operator on the Hilbert space $\mathcal{H}$ characterizing the energy of the physical system. If the Hamilton operator H in (3.3) is not time-dependent, and the initial state $|\psi(0)\rangle$ is an eigenvector of H to the eigenvalue $\lambda \in \mathbb{R}$, (3.3) has the close-form solution

$$|\psi(t)\rangle = \exp\left\{-\frac{i}{\hbar}\frac{t}{\lambda}\right\}|\psi(0)\rangle\,, \tag{3.4}$$

such that $|\psi(t)\rangle$ remains in the same state modulo a phase.

Adiabatic Quantum Computing. AQC solves quadratic unconstrained binary optimization problems (QUBO) on quantum hardware relying on the *adiabatic quantum theorem* [44]. Assume one faces the following QUBO:

$$\underset{\mathbf{s}\in\{-1,1\}^n}{\arg\min} \quad \mathbf{s}^T Q\mathbf{s} + \mathbf{q}^T\mathbf{s}, \tag{3.5}$$

with an $n \times n$-matrix Q and an n-dimensional vector $\mathbf{q}$. Furthermore, assume one is able to set up a quantum system with a Hamilton operator $H(0) =: H_B$ in a state $|\psi(0)\rangle$ that is an eigenvector to the smallest eigenvalue of H_B, and one constructs a second Hamilton operator H_P whose eigenvector to the smallest eigenvalue is a solution to (3.5). Then, [44] dictates that the time evolution of (3.3) with

$$H(t) = \frac{t}{\tau}H_P + \Big(1 - \frac{t}{\tau}\Big)H_B, \tag{3.6}$$

for $t \in [0, \tau]$ with large τ to slowly transition between H_B and H_P, remains in the eigenvector with the smallest eigenvalue (provided that it remains unique and non-degenerate), so that measuring the corresponding state $|\psi(\tau)\rangle$ at $t = \tau$ yields the solution to (3.5). For the cases we investigate in the paper, H_P is a diagonal matrix and for $|\psi(0)\rangle$, all components are equal in the computational basis. To experimentally realize an evolution with the Hamiltonian H_P, one needs to provide couplings between the qubits in form of the matrix Q and the biases as the vector q upon (3.5).

Of course, an infinitely slow transition between the two Hamilton operators, *i.e.,* $\tau \to \infty$, is impossible (and would eliminate any computational advantages). Luckily, it has been shown that suitable finite choices of τ to remain in the adiabatic case depend on the *spectral gap* of the problem, *i.e.,* the difference between the smallest and second lowest eigenvalue. For more details on how to estimate τ and spectral gap analysis, see [12, 134].

D-Wave 2000Q. D-Wave 2000Q is an AQC-er with 2048 qubits arranged on a *Chimera* graph, *i.e.,* a 16×16 set of cells with eight qubits each. The qubits inside a cell are fully connected, whereas there are only eight connections to qubits from two other neighbouring cells [70]. The computation always starts with the initialization state $|\psi(0)\rangle = |+\rangle^{\otimes n}$, where n is the number of physical qubits in the problem mapping to the Chimera graph, also known as *minor embedding*. Logical problem qubits are often mapped to chains of physical qubits, due to the restriction on the physical qubits connectivity. The D-Wave compiler performs the minor embedding automatically, and the programmer loads couplings Q and biases q for the target problem. Moreover, additional settings can be provided such as the chain strength, annealing time and the annealing path.

The default duration of an anneal is $20\mu s$. To solve a problem, usually multiple anneals

of the same problem instance are performed, since there is no guarantee (due to the physical characteristics of current quantum hardware) that D-Wave will return optimal solutions.

The couplers are realized with loops of niobium. During the annealing, the processor imposes magnetic fluxes on the couplers corresponding to the provided Q matrix.

3.4 Related Work

Our approach relates to multiple method classes proposed in the literature so far. In this section, we review the most related categories, *i.e.,* computer vision methods using quantum hardware including image matching and encoding of permutation matrices, as well as classical solutions to graph matching problems.

Quantum Computing in Computer Vision. Promising potential applications of quantum computers range from data fitting [267, 61] and image recognition [194, 193, 206] to training artificial neural networks [8, 143]. Classification-related problems were addressed in [45], to enhance the detection of vegetation zones in aerial images on D-Wave, in [206] to learn facial features and reproduce facial image collections, and in [195] which proposes a dictionary learning method for image classification. [273] covers various low-level quantum image processing topics such as quantum image encryption and segmentation. However, it does not address integration of permutation matrix constraints. To the best of our knowledge, [111] is the first work introducing quantum computing for computer vision at a major vision conference. The method finds a rotation that aligns two point sets and discretizes the space of affine transformation matrices to formulate the problem as (3.5).

Quantum Computing for Matching Problems. [136] finds a small image embedded in a larger one by adapting the Grover's algorithm [117]. Neven *et al.* [194] propose to match two images with AQC-ing. Their mapping requires finding the maximum independent vertex set of a conflict graph without a focus on permutation constraints.

Examples of solving quadratic assignment problems with AQC-ing are [247, 63]. Stollenwerk *et al.* [247] solve the flight gate assignment problem by using qubits to indicate whether resources (*e.g.,* flights) are assigned to facilities (*e.g.,* gates). The resulting matrix is indexed over the resources and facilities, and the quadratic constraints are converted to weights favoring valid configurations in the solution space. Choo [63] investigates several ways to map quadratic assignment problems to QUBO. Their analysis leads mainly to a quadratic penalty terms. They also propose a decomposition approach to mitigate the challenges.

[107, 279] addresses graph isomorphism problem with AQC-ing. In [107], permutation matrix constraints are enforced with a penalty term in the Hamiltonian. They first write the permutation in a table and then represent the entries in the binary system. Zick *et al.* [279] allocate a QUBO variable for every possible pair of vertices of the same degree and solve the problem with AQC-ing, but do not use a binary representation of the permutation table. In this paper, we investigate a different new way of injecting permutation matrix constraints.

Classical Related Methods. In terms of classical approaches to solve matching problems, it is well-known that problems of the form (3.1) can be solved exactly if the costs are linear ($W = 0$) using the Hungarian method [156] or the auction algorithm [36]. For truly quadratic costs, relaxations based on message passing [152] and Lagrange duality [251, 257], as well as general convex approximations [234, 9, 33, 98, 104, 49] have been proposed. Path following

[276, 278, 135] and believe propagation [14] are popular heuristics to tackle (3.1) directly, and we refer to [209] for an overview of combinatorial approaches to the QAP.

A magnitude of the methods is inspired by quantum mechanics while not being meant for quantum hardware including *quantum cuts* for image segmentation [23], *wave kernel signatures* [22] for mesh matching and *genetic quantum algorithms* [141] for the knapsack problem. None of these methods operates on permutation matrices. A related probabilistic physics-based technique which can be applied to problems on graphs is simulated annealing [149]. In [126], (3.1) arises for finding the maximal pair of subgraphs between two graphs. The vertices of these graphs represent the interest points in the respective images one wants to match.

3.5 Minimization of Quadratic Objectives over Permutation Matrices

In this section, we derive three variants of our QGM approach, *i.e.,* baseline variant (Sec 3.5), the variant with row-wise penalties which reduces the penalty-based influence on the problem in overall (Sec. 3.5.2) and the variant with eliminated variables (Sec. 3.5.3).

3.5.1 Baseline Quantum Graph Matching

To rephrase (3.1) in the form of (3.5), we have to do two conversions: 1) Switch from $x_i \in \{0,1\}$ to $s_i \in \{-1,1\}$, and 2) Switch from the constrained problem over permutation matrices to an unconstrained problem. The first one is straightforward by substituting $s_i = 2x_i - 1$, yielding

$$Q = \frac{1}{4}W, \qquad q = \frac{1}{2}(\mathbb{1}^T W + \mathbf{c}), \tag{3.7}$$

between the costs associated with (3.1) and (3.5), respectively, up to a constant (assuming without restriction of generality that W is symmetric, and denoting by $\mathbb{1}$ a vector of ones).

The second conversion is less trivial. First of all, note that the set $\mathbb{P}_n$ of permutation matrices can be written as

$$\begin{aligned} \mathbb{P}_n = \{&X \in \mathbb{R}^{n\times n} \mid X_{ij} \in \{0,1\}, \\ &\sum_i X_{ij} = 1 \ \forall j, \ \sum_j X_{ij} = 1 \ \forall i\}. \end{aligned} \tag{3.8}$$

Note that – besides the binary constraint addressed with the above transformation – $\mathbb{P}_n$ is now characterized by linear equality constraints. Thus, after the vectorization of $X \in \mathbb{P}_n$ to $\mathbf{x} \in \mathbb{R}^{n^2}$, one can phrase such constraints in matrix-vector form as $A\mathbf{x} = \mathbf{b}$, for a suitable matrix A and a vector $\mathbf{b}$. For $n = 2$, as an example, A and $\mathbf{b}$ are given by:

$$A = \begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{pmatrix}, \quad \mathbf{b} = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}. \tag{3.9}$$

For higher dimensions, b still contains only ones. The matrix A for arbitrary n can be described

with the formula:

$$A = \begin{pmatrix} \mathrm{Id} \otimes \mathbb{1}^{\mathrm{T}} \\ \mathbb{1}^{\mathrm{T}} \otimes \mathrm{Id} \end{pmatrix}, \tag{3.10}$$

where Id is the $n-$dimensional identity matrix. It is well known (see *e.g.,* [151]) that linear equality constraints can be incorporated in QUBO by adding a penalty term to the objective, *i.e.,*

$$\begin{aligned} &\underset{\{\mathbf{x}\in\{0,1\}^{n^2} \mid \quad A\mathbf{x}=\mathbf{b}\}}{\arg\min} \quad \mathbf{x}^{\mathrm{T}} W \mathbf{x} + \mathbf{c}^T \mathbf{x} \\ &= \underset{\mathbf{x}\in\{0,1\}^{n^2}}{\arg\min} \quad \mathbf{x}^{\mathrm{T}} W \mathbf{x} + \mathbf{c}^T \mathbf{x} + \lambda ||A\mathbf{x} - \mathbf{b}||^2, \end{aligned} \tag{3.11}$$

for sufficiently large λ. For the sake of completeness, we provide a proof of this statement, the specific appearance of A and $\mathbf{b}$, and the lower bound

$$\lambda > \lambda^0 := \frac{1}{2} \left(\sum_{i,j=1}^{N} |W_{i,j}| + \sum_{i=1}^{N} |c_i| \right) \tag{3.12}$$

for (3.11) to provably hold in the supplementary material. After rewriting the second line of (3.11) to

$$\underset{\mathbf{x}\in\{0,1\}^{n^2}}{\arg\min} \quad \mathbf{x}^{\mathrm{T}} (W + \lambda A^T A) \mathbf{x} + (\mathbf{c}^T - 2\lambda \mathbf{b}^T A) \mathbf{x}, \tag{3.13}$$

we arrive at a form of the minimization problem (3.1) to which quantum computing is directly applicable.

Interestingly, applying quantum computing approaches to (3.13) often does not yield satisfactory results as we will illustrate in more detail in Sec. 3.6. We conjecture that this behavior is largely due to a significant reduction of the *spectral gap* of the coupling matrix $Q + \lambda \frac{1}{4} A^T A$ with increasing λ (after the global scale-normalization). As discussed in Sec. 3.3, the spectral gap *i.e.*, the smallest difference between the smallest and the second lowest eigenvalues of the Hamiltonian (3.6) over time, has a crucial influence on how slow the transition needs to be and how likely the system leaves its ground state during the time evolution.

The penalty in (3.13) is not the only reformulation of the constrained QAP as an unconstrained QUBO. We next derive two alternate formulations (Secs. 3.5.2–3.5.3) and evaluate them in terms of their resulting spectral gaps as well as behavior on real quantum hardware (Sec. 3.6).

3.5.2 The Formulation with a Row-Wise Penalty

As large values of λ seem to cause problems, we aim at reducing the penalty-based changes of the problem as little as possible by considering row-wise penalty parameters λ_i for enforcing each equation $(A\mathbf{x})_i = b_i$ separately, *i.e.,*

$$\min_{\mathbf{x}\in\{0,1\}^{n^2}} \quad \mathbf{x}^{\mathrm{T}} W \mathbf{x} + \mathbf{c}^T \mathbf{x} + \sum_{i=1}^{2n} \lambda_i ((A\mathbf{x})_i - b_i)^2. \tag{3.14}$$

While a change of variables from x to s along with the choice $\lambda_i = \lambda > \lambda^0$ according to (3.12) is obviously sufficient and reduces (3.21) to (3.13), the challenge is to find small *computable lower bounds* for λ_i, for which the unconstrained problem still *provably* coincides with the constrained problem (3.1).

Proposition 3.1. *The minimizers of* (3.5) *and* (3.21) *coincide provided that*

$$\lambda_i > \lambda_i^0 := D_{\mathcal{J}_i} + \frac{1}{2} D_{\{1,\dots,n^2\}}, \tag{3.15}$$

where $\mathcal{J}_i$ *denotes the indices that belong to a column or a row enumerated by* i *in* (3.21), $D_{\{1,\dots,n^2\}} = D_{\mathcal{J}}$ *for* $\mathcal{J} = \{1, \dots, n^2\}$ *and*

$$D_{\mathcal{J}} := \max_{k \in \mathcal{J}} \Big(\sum_i |(W_{k,i} + W_{i,k})| + |W_{k,k}| + |c_k| \Big). \tag{3.16}$$

3.5.3 Inserted Formulation with Eliminated Variables

In continuous optimization, linear equality constraints are frequently used to eliminate variables and reduce the complexity of the underlying problem. Even in our discrete setting, the sum-to-one constraints of permutation matrices in (3.8) naturally allow expressing $X_{1,j} = 1 - \sum_{i=2}^n X_{i,j}$ for all j as well as $X_{i,1} = 1 - \sum_{j=2}^n X_{i,j}$ as long as one ensures that the variables to be replaced remain in $\{0,1\}$. In other words:

Lemma 3.2. *The set in* $\mathbb{P}_n$ *of* (3.8) *can be written as*

$$\begin{aligned}
\mathbb{P}_n = \{ & \\
& \begin{pmatrix} 2 - n + \sum_{i,j=2}^n x_{i,j} & 1 - \sum_{i=2}^n x_{i,2} & \dots & 1 - \sum_{i=2}^n x_{i,n} \\ 1 - \sum_{i=2}^n x_{2,i} & x_{2,2} & \dots & x_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ 1 - \sum_{i=2}^n x_{n,i} & x_{n,2} & \dots & x_{n,n} \end{pmatrix} \\
& \in \mathbb{R}^{n \times n} | \forall i \in \{2, \dots, n\}, j \in \{2, \dots, n\} \; x_{i,j} \in \{0,1\}, \\
& \sum_{i,j=2}^n x_{i,j} \in \{n-2, n-1\}, \\
& \forall j, i, k \in \{2, \dots, n\}, \;\; i \neq k, \;\; x_{j,i} x_{j,k} = 0 \wedge x_{i,j} x_{k,j} = 0 \}.
\end{aligned}$$

This reformulation of the constraint set allows us to reduce the amount of needed qubits from n^2 to $(n-1)^2$ by incorporating all remaining constraints via penalties as stated in the following proposition.

Proposition 3.3. *The problem* (3.1) *has the same solution as the following QUBO for sufficiently large* λ_1 *and* λ_2*:*

$$\min_{\mathbf{x} \in \{0,1\}^{(n-1)^2}} \quad \mathbf{x}^T \tilde{W} \mathbf{x} + \tilde{\mathbf{c}}^T \mathbf{x} + \lambda_1 \sum_{j,k=0}^{(n-1)^2} \sum_{j \neq k} \chi(j,k) x_j x_k$$

$$+\lambda_2\left(\sum_{i=1}^{(n-1)^2} x_i-(n-1)\right)\left(\sum_{i=1}^{(n-1)^2} x_i-(n-2)\right)$$

$$\chi(j,k):=\begin{cases}1 & \text{for } \lfloor\frac{j}{n-1}\rfloor=\lfloor\frac{k}{n-1}\rfloor \\ & \vee j-(n-1)\lfloor\frac{j}{n-1}\rfloor=k-(n-1)\lfloor\frac{k}{n-1}\rfloor\,, \\ 0 & \text{otherwise}\end{cases}$$

where the specific definitions of the reduced $(n-1)\times(n-1)$ *matrix* $\tilde{W}$*, the vector* $\tilde{\mathbf{c}}\in\mathbb{R}^{n-1}$ *and the precise lower bounds for* λ_1 *and* λ_2 *are provided in the supplement.*

The reduction in the number of required qubits is a clear advantage of the above *inserted* formulation, especially in the current early stage of AQC-ers, when the number of supported logical qubits is limited [70]. On the other hand, the asymptotic difference between the n^2 and $(n-1)^2$ (of the inserted formulation) vanishes, such that their spectral gaps, and, more importantly, their performance on real AQC-ers are the deciding factors we evaluate in the next section.

3.6 Experimental Results

We next test our methods for quantum graph matching on random problem instances in Sec. 3.6.1 and real data in Sec. 3.6.3. We also analyze in Sec. 3.6.2 the influence of experimental errors in the coupling parameters to the measured results. All in all, we investigate the following questions:

- Do the three equivalent reformulations of (3.1), *i.e.,* the baseline, the row-wise penalty and the inserted formulations lead to different spectral gaps?
- Do larger spectral gaps improve AQC-ing results?
- Do quantum computing approaches provide advantages over classical relaxation methods for solving (3.1)?
- Do AQC-ers – as well as their simulations, *i.e.,* numerical solutions of (3.3) – solve our problem?

We prepare the weight matrix W on a classical computer, and access D-Wave 2000Q remotely in the cloud via *Ocean Tools Leap 2* [72]. In all experiments, we report the solution distributions and analyse the probability and the energy of the most frequent solution. The ground-truth permutations are calculated by brute force and compared qualitatively with the expected outcomes on real data. Minor embedding of QGM results in a fully connected logical graph of qubits. For $n=3$, the number of physical qubits in the minor embedding amounts to $\approx$27.

3.6.1 Evaluation on Random Problem Instances

As the first step, we create random problems of the form (3.1) by drawing all entries in W and c uniformly from $[-1,1]$. Those are representative for a broad class of problems arising in 3D vision such as mesh and point cloud matching under isometry, and rigid point set alignment. We compute the spectral gap for each of our three formulations via an iterative Lanczos method [157], since this calculation requires eigenvalue decomposition of large quadratic matrices with the dimension 2^{n^2}. Fig. 3.2 shows the spectral gaps of all three formulations averaged over ten

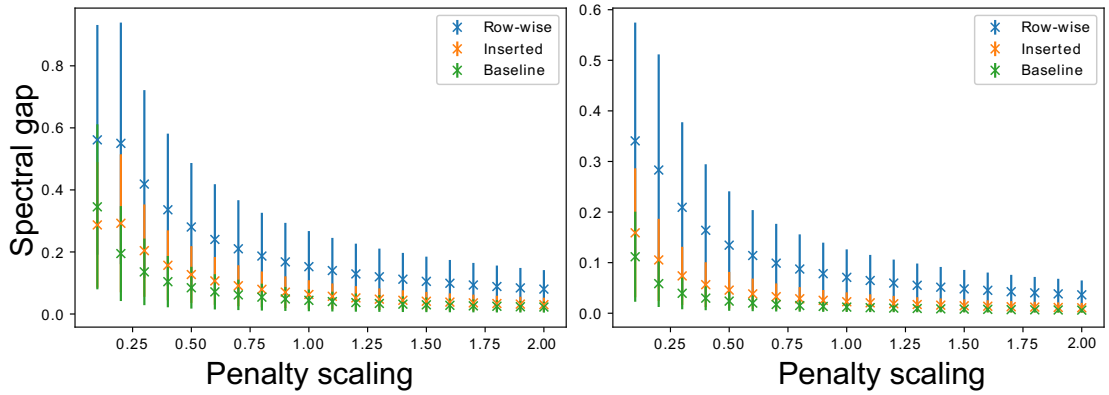


Figure 3.2: Illustrating the spectral gap for our baseline, the row-wise penalty and inserted formulation as a function of the global scaling of the penalty term for $n = 3$ (left) and $n = 4$ (right). As we see, the row-wise penalty leads to the largest spectral gap which is favorable for AQC-ing.

different problem instances in dimensions $n = 3$ and $n = 4$, when the strength of the overall penalty is varied by a global scale factor (x-axis). A scale factor of 1 refers to the setting where the constrained and unconstrained problems are provably equivalent according to our derivations in Sec. 3.5. As we see, the spectral gap reduces significantly for all formulations and instances with the increasing strength of the constraint penalty. Among the three formulations, the row-wise penalty consistently results in the largest spectral gap which we expect to be favorable for real and simulated AQC-ing.

Fig. 3.3 shows the different results of our three reformulations run on D-Wave, and the recent relaxation-based algorithm [33] for $n \in \{3, 4\}$ over ten instances of (3.1). For each instance, we compute the global optimizer by brute force and normalize (shift) the energy by $-f_{opt}$ for each method.

For the AQC-ing approach, we use 500 annealing cycles on the D-Wave 2000Q and select the most frequent solution among all annealing cycles as our final solution. On D-Wave, logical problem qubits are mapped to multiple physical qubits by chaining. For $n = 4$, we set the chain strength as the maximal element of the q vector in (3.7). This leads to an automatic adaption of the chain strength to the problem instance. For $n = 3$, this does not lead to further improvements over empirical fixed values of the chain strength. Here, we have the chain strengths $20, 30$ and 35 for the inserted, the row-wise and the baseline method, respectively. Up to a $100\mu s$ break, we insert in the middle of the annealing path, the annealing time τ is $80\mu s$ for $n = 4$. For $n = 3$, $\tau = 20\mu s$ already yields desired result.

The simulation with QuTiP [137] showed that increasing the τ yields a narrower distribution. For $n = 3$, the simulation can find the correct optimum as the highest peak regardless of the method variant. The situation changes for $n = 4$ and the simulation with the *baseline* formulation does not yield satisfactory results. The method with inserted constraints, on the other hand, finds the minimum every time.

Table 3.1 summarizes the mean normalized energies over ten instances for $n = 3$ and $n = 4$,

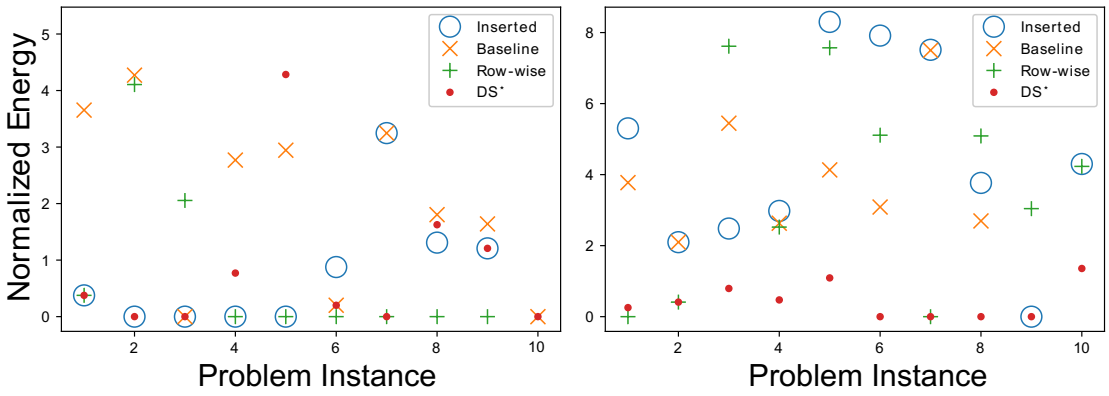


Figure 3.3: Comparing quantum computing on DWave 2000Q for our three reformulations and the relaxation-based method DS* [33] for computing 3×3 (left) and 4×4 permutations (right) on ten random problem instances.

n	inserted	baseline	row-wise	DS* [33]	worst permutation
3	1.49	2.12	2.27	0.85	4.62
4	5.68	7.37	7.40	0.43	10.23

Table 3.1: Average energies for $n = 3, 4$.

with the averages computed so that 0 is the best achievable value. Note that if AQC-ing yields a state that is not a valid permutation, the worst permutation is used for averaging. Compared to the simulation, where we were always able to find the global solution with the inserted or the row-wise method for large enough τ, AQC-ing on quantum hardware is significantly more difficult. As we can see in Fig. 3.3-(left), the most probable states have the lowest energy for the inserted and the row-wise method, and the results are comparable to DS* [33] in $n = 3$. For $n = 4$ the DS* performs clearly the best.

Interestingly, for $n = 3$, the row-wise penalties were able to substantiate their advantages with respect to the spectral gap in practice, yielding several problem instances on which it outperformed DS* and being on par with it on average. While the dimension of the underlying problems is of course small (for $n = 3$, there only exist six different permutation matrices), we consider these results to be very promising, as they illustrate the potential of AQC-ing beyond pure simulations with future generation of AQC-ers.

3.6.2 Ablative Study on the Coupling Parameters

On D-Wave, the input Q and $\mathbf{q}$ values are scaled so that $-1 < Q_{i,j} < 1$ and $-2 < q_i < 2$. Additionally, there are multiple further error sources for the couplings and biases [2]. Therefore, it can happen that the input $Q_{i,j} = 0.5$ is realized as 0.51, for example. As we saw in Sec. 3.5, Eqs. (3.11) and (3.21), the Hamiltonian has a part that depends on the problem instance W, c directly and includes the regularization term which only depends on W, c in the penalty parameters.

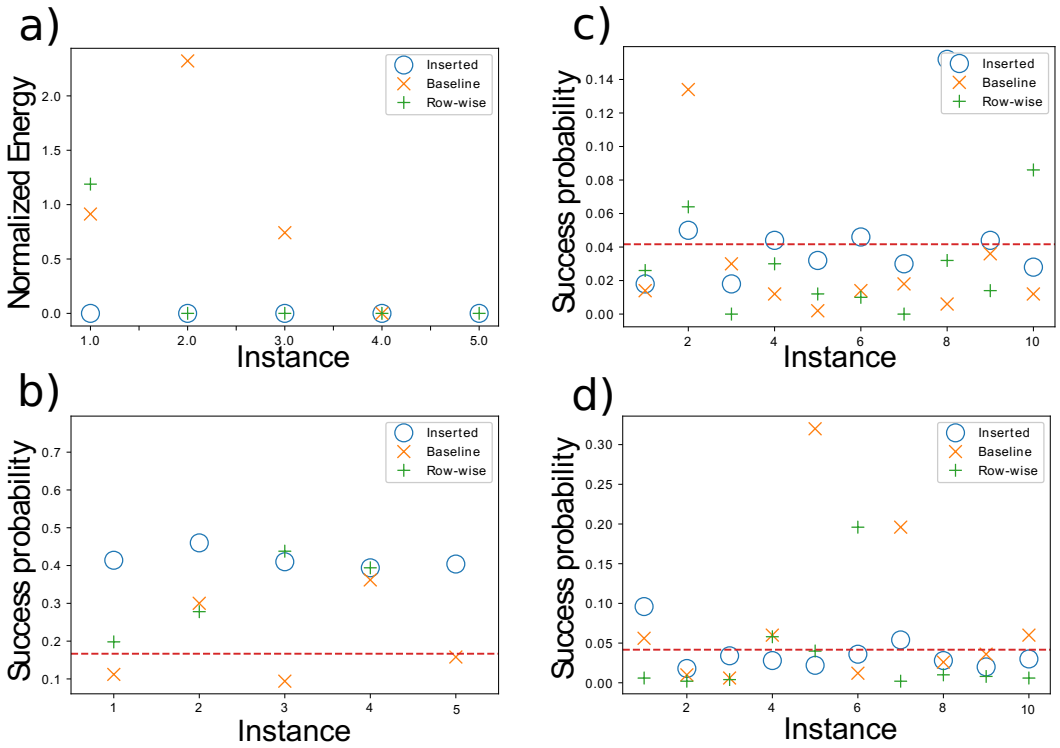


Figure 3.4: Problem instances, where multiple elements are randomly chosen to be zero. In **a)** and **b)**, the problem instances are for $n = 3$. Here, less than half of the entries in W and c are randomly set zero. **c)** and **d)** show the success probability for $n = 4$. For **c)**, exactly half of the entries are chosen zero and for **d)**, exactly 3/4 are chosen zero.

The regularization part yields the same energy for every valid permutation. Let Q_{reg} and q_{reg} be the couplings and biases that stem from the regularization term.

If permutations only marginally differ in the qubit couplings, it is difficult to distinguish between them during a quantum anneal. This suggests that we have to choose the penalties as small as possible while already delimiting from random binary strings. Assuming that all values in W are of the same order of magnitude, λ would be n^4 times bigger than W values for the baseline method. Our calculations also confirm that for $n = 4$, the maximal value of c that is due to the penalty, is 500 times larger than the maximal value of c in the problem itself.

One easy way to have a larger (more beneficial) ratio between the range of the original W values and the range value in the penalty term is to consider problems with W entries randomly set to zero. The connectivity should stay the same, due to the regularization. For $n = 4$, we either set a half or $\frac{3}{4}$ of the entries of the problem instance to zero. The results in Fig. 3.4 show that for the case $n = 4$, random guessing is still better most of the time.

3.6.3 Experiments with Real Data

We now solve several real problems of matching two graphs originating from two 3D shapes deformed in a near-isometric way. Assuming that we have an edge-weight or a notion of distance

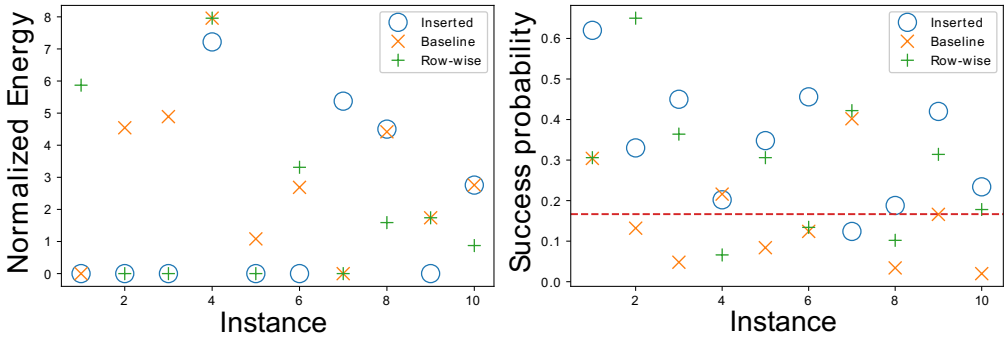


Figure 3.5: Performance of the quantum annealer in the shape matching problem for $n = 3$.

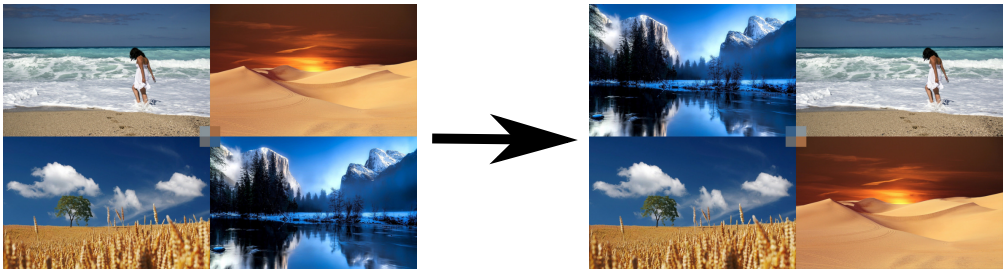


Figure 3.6: Example of image matching for $n = 4$.

$d_1(v_i^1, v_k^1)$ between nodes v_i^1 and v_k^1 of our first graph, and, similarly, a distance $d_2(v_j^2, v_l^2)$ between nodes v_j^2 and v_l^2 of our second graph, a common choice for the quadratic costs in (3.1) is

$$\sum_{i,j,k,l} X_{i,j} X_{k,l} |d_1(v_i^1, v_k^1) - d_2(v_j^2, v_l^2)|. \tag{3.17}$$

The above formulation assures that if v_i^1 is matched to v_j^2 and v_k^1 is matched to v_l^2, then the distance between v_i^1 and v_k^1 is similar to the distance between v_j^2 and v_l^2.

We consider different instances of (3.17): In our first shape matching example, we aim to identify the corresponding points on two differently deformed instances of the same shape. In this case, d_1 and d_2 denote the geodesic distance on the respective shapes, and we introduce an additional linear term based on the Euclidean distance of the points after an initial rigid registration following [59]. Fig. 3.1 visualizes the matching of the two shapes along with the histogram we obtained from 500 runs of this problem instance on the D-Wave 2000Q. See Fig. 3.5 for the performance comparison of different QGM variants. We observe that the results with real 3D shapes match coarsely the distributions of solutions to experiments with random instances in Sec. 3.6.1.

Fig. 3.6 illustrates another application of costs (3.17) for sorting images according to their mean color as considered in [98], see Fig. 3.6 for the visualization. In this setting, d_1 represents the difference in the mean image color, and d_2 is the Euclidean distance on the 2×2 image grid. The mean image color is sampled from the middle region of each image. As we can see, after the sorting on the right, the images are ordered from predominant blue on the top left (mountains

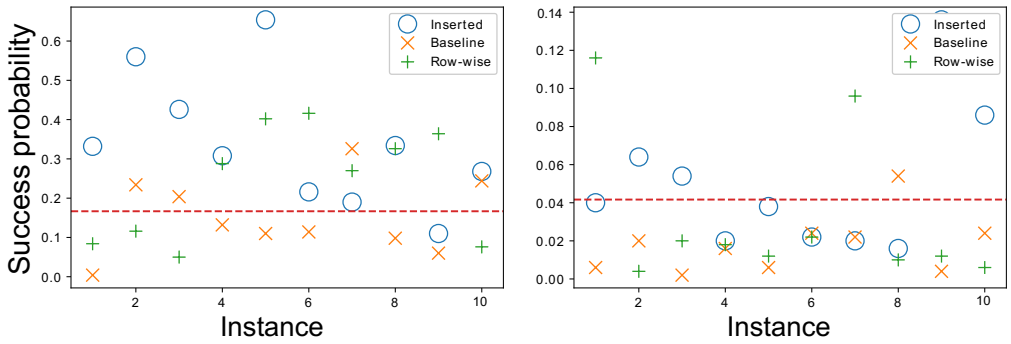


Figure 3.7: The probability to measure the optimum in one run compared to random guessing of permutations for $n = 3$ (left) and $n = 4$ (right). The problem instances are dense, *without* zero entries.

with a lake), to the predominant yellow (desert with the sunset). This problem, likewise, has multiple possible minimizers, and the quantum annealer finds the optimum more often than the other (non-optimal) states.

3.7 Further Observations and Limitations

We were able to successfully confirm in our experiments the validity of our QGM approach for $n = 3$. On a wide range of randomly generated problem instances, our method outputs optimal valid permutations with high probability. On the current D-Wave hardware, the results for $n = 4$ are significantly worse, *i.e.,* neither do we obtain a conclusive best choice between the three different formulations, nor do we get near the performance of the DS* heuristic [33]. Fig. 3.7 shows the probabilities of obtaining the optimal solution based on 500 anneals per instance. For most instances, the different formulations performed even worse than random guessing, because they can also output vectors that do not correspond to permutations. These results cannot be due to chain breaks, because the D-Wave compiler did not report any. Another interesting quantity is the length of the chain. For $n = 2$, our chains have maximal length of two, whereas for $n = 3$, there are also chains of length three and four. For $n = 4$ the chains have length five and six.

Remark on IBM Q Experience. We also run tests via the IBM Q Experience [1]. They offer not an AQC-er but a circuit-based universal quantum computer, where the algorithm can be viewed as a circuit consisting of so-called *quantum gates*. To test our method on it, we discretize our Hamiltonian with constant time steps and apply trotterization [197] for the decomposition into quantum gates. For higher-dimensional problems, the number of gates rises, which leads to errors. We thus are only able to successfully realize an adiabatic transition for two qubits. By inserting two equality restrictions, we can optimize over 2×2 permutations. We provide more details on this alternative direction in the supplementary material.

3.8 Conclusions

The quadratic assignment problem which we tackle in this paper is a hard problem to be mapped and solved on a real quantum annealer, and we make a further successful step to solve it on real AQC-ers. The difficulties come from the high demand in connectivity between the qubits and the strong scaling in the qubit number ($n \log(n)$ in the best case). We achieve expected results for 3×3 permutations and demonstrate the potential of the new quantum method for solving quadratic assignment problems in 3D vision, whereas our formulation can be applied to problems in different dimensions which is demonstrated experimentally. The spectral gap analysis shows that it greatly matters how the problem is mapped to a QUBO and subsequently solved by an AQC-er. We expect a significant impact of the future quantum technology on 3D vision, while the accessibility of modern AQC-ers for research purposes allows us to lay the foundation already today. In future work, we will investigate other structures such as point clouds with outliers, more nodes and partial matches.

Supplementary Material. Our supplement contains the proofs of Lemma 3.2 and Prop. 3.1 as well as more experimental results.

Acknowledgement. This work was partially supported by the ERC Consolidator Grant 4DReply (770784).

3.9 Supplementary Material for Adiabatic Quantum Graph Matching with Permutation Matrix Constraints

This supplementary material provides details on the derivations of the Quantum Graph Matching (QGM) approach as well as more experimental results. The proofs of Lemma (3.2) and Prop. (3.1) of the main matter can be found in Sec. 3.10. Sec. 3.11 complements the description of our experiments with the quantum computers, accessible via IBM Quantum Experience. Next, we show additional results of QGM for point cloud alignment in Sec. 3.12 and nearly isometric non-rigid shape registration in Sec. 3.13, followed by a discussion of the minor embedding and negative results in Secs. 3.14 and 3.15, respectively. Note that D-Wave provides one minute of QPU time on their annealers for research purposes per month. We obtain three minutes in total during the current project and use 80% of this time for our experiments. One minute of time on the quantum annealer can suffice to run hundreds of problems.

3.10 Proofs for Section 3.5

In order to ensure that we can incorporate a constraint of the form $A\mathbf{x} = \mathbf{b}$ as a penalty, the corresponding penalty parameter λ has to be chosen such that

$$\min_{\mathbf{x}\in\{0,1\}^n, A\mathbf{x}=\mathbf{b}} \mathbf{x}^T W \mathbf{x} < \min_{\mathbf{x}\in\{0,1\}^n, A\mathbf{x}\neq\mathbf{b}} \mathbf{x}^T W \mathbf{x} + \lambda \|A\mathbf{x} - \mathbf{b}\|^2. \tag{3.18}$$

A naive estimate for a sufficiently large λ is obtained by using

$$\min_{\mathbf{x}\in\{0,1\}^n, A\mathbf{x}=\mathbf{b}} \mathbf{x}^T W \mathbf{x} \leq \sum_{i,j} \max\{0, W_{i,j}\} \tag{3.19}$$

and

$$\begin{aligned}
&\min_{\mathbf{x}\in\{0,1\}^n, A\mathbf{x}\neq\mathbf{b}} \mathbf{x}^T W\mathbf{x} + \lambda\|A\mathbf{x}-\mathbf{b}\|^2 \\
\geq\ &\min_{\mathbf{x}\in\{0,1\}^n, A\mathbf{x}\neq\mathbf{b}} \mathbf{x}^T W\mathbf{x} + \lambda \min_{\mathbf{x}\in\{0,1\}^n, A\mathbf{x}\neq\mathbf{b}} \|A\mathbf{x}-\mathbf{b}\|^2 \\
\geq\ &\min_{\mathbf{x}\in\{0,1\}^n} \mathbf{x}^T W\mathbf{x} + \lambda \underbrace{\min_{s\in\{-1,1\}^n, A\mathbf{x}\neq\mathbf{b}} \|A\mathbf{x}-\mathbf{b}\|^2}_{=:d} \\
\geq\ &-\sum_{i,j} \max\{0, -W_{i,j}\} + \lambda d.
\end{aligned} \tag{3.20}$$

Thus a sufficient upper bound for λ is given by

$$\begin{aligned}
\lambda &> \frac{1}{d}\sum_{i,j}(\max\{0,-W_{i,j}\} + \max\{0,W_{i,j}\}) \\
&= \frac{1}{d}\sum_{i,j}|W_{i,j}|.
\end{aligned}$$

The influence of the linear term $\mathbf{c}^{\mathrm{T}}\mathbf{x}$ can be calculated in the same way. Alternatively $\mathbf{c}$ can be added to the diagonal of W, since $x_i^2 = x_i$ for $x_i \in \{0,1\}$. The fact that $d = 2$ can be proven by the following reasoning. Consider that there is a violation for the rows so that one row does not sum up to one and $\sum_{k \text{ belongs to a row}}(Ax-b)_k^2 = 1$. Since the sum over all rows of a matrix is the same as the sum over all columns, one constraint for the columns is violated as well. Since the same argument can be made if we switch rows and columns d has to be 2.

3.10.1 Proof of Proposition (3.1)

Better estimates can be obtained by exploiting more about the specific structure of the constraint. We aim to formulate the optimization problem as an unconstrained optimization problem with row-wise penalty parameters:

$$\min_{\mathbf{x}\in\{0,1\}^{n^2}} \quad \mathbf{x}^{\mathrm{T}} W\mathbf{x} + \mathbf{c}^T\mathbf{x} + \sum_{i=1}^{2n} \lambda_i((A\mathbf{x})_i - b_i)^2. \tag{3.21}$$

One can solve the following optimization problems

$$D_{\mathcal{J}_j} = \max_{k\in\mathcal{J}_j}(\sum_i |(W_{k,i} + W_{i,k})| + |W_{k,k}| + |c_k|) \tag{3.22}$$

for every row or column $\mathcal{J}_j$.

The following bound for λ_j can be used to obtain an unconstrained optimization problem as in (3.21):

$$\lambda_j = \max_{k\in\mathcal{J}_j}(\sum_i |(W_{k,i} + W_{i,k})| + |W_{k,k}| + |c_k|) \tag{3.23}$$

$$+\frac{1}{2}(\max_k \sum_i |(W_{k,i}+W_{i,k})|+|W_{k,k}|+|c_k|) \tag{3.24}$$

$$= D_{\mathcal{J}_j} + \frac{1}{2}D_{\{1,\dots,n^2\}} \tag{3.25}$$

Proof. Let x be an arbitrary element in $\{0,1\}^{n^2}$. We want to show that for the above choice of λ_j

$$\exists \mathbf{p} \in \mathrm{vec}(\mathbb{P}_n) : \mathbf{p}^{\mathrm{T}} W \mathbf{p} + \mathbf{c}^T \mathbf{p} \tag{3.26}$$

$$\leq \mathbf{x}^{\mathrm{T}} W \mathbf{x} + \mathbf{c}^T \mathbf{x} + \sum_{i=1}^{2n} \lambda_i ((A\mathbf{x})_i - b_i)^2 \tag{3.27}$$

For the matrix X, which fulfills $x = \mathrm{vec}(X)$, we can construct sets with the property:

$$\begin{aligned} (i,j) \in I \Rightarrow X_{i,j} = 1 \quad &\wedge \quad \forall k \in \{1,...,n\} \setminus \{j\} (i,k) \notin I \\ &\wedge \quad \forall k \in \{1,...,n\} \setminus \{j\} (k,j) \notin I, \end{aligned} \tag{3.28}$$

We name one of these sets that has the maximal possible number of elements $I_{\max}$. The permutation matrix P with $\mathbf{p} = \mathrm{vec}(P)$ that we want to construct can be any permutation with ones placed at the positions in $I_{\max}$. We can get from X to P by first erasing all ones that are not in the positions $I_{\max}$ and then adding $n - |I_{\max}|$ ones.

Consider the set of matrices $(X^{(k)})_{0 \leq k \leq H}$ with:

$$\begin{aligned} X^{(0)} &= X \\ X^{(H)} &= P \\ ||X^{(k)} - X^{(k-1)}|| &= 1 \end{aligned} \tag{3.29}$$

These matrices can be constructed if we start from P and erase successively all ones that are not in $I_{\max}$. After that we can insert ones that have an index in common with an element in $I_{\max}$. This set we call B.

Inserting or erasing a one at the j-th column yields maximally to an energy difference of

$$D_{\mathcal{C}_j} = \max_{k \in \mathcal{C}_j} (\sum_i |(W_{k,i}+W_{i,k})| + |W_{k,k}| + |c_k|), \tag{3.30}$$

where $\mathcal{C}_j$ describe the indizes that belong to the j-th collumn and analogously $\mathcal{R}_j$ describe the indizes that belong to the j-th row. We define f as:

$$f(Y) = \mathbf{y}^{\mathrm{T}} W \mathbf{y} + \mathbf{c}^T \mathbf{y} \tag{3.31}$$

with $\mathbf{y} = \text{vec}(Y)$. To prove (3.27) we use the principle of a telescope sum:

$$\begin{aligned} & f(P) - f(X) \\ &= \sum_{k=0}^{H-1} f(X^{(k+1)}) - f(X^{(k)}) \leq \sum_{k=0}^{H-1} \left| f(X^{(k+1)}) - f(X^{(k)}) \right| \\ &= \sum_{\{k \in \{0,\dots,H-1\} | \text{pos } (X^{(k+1)} - X^{(k)}) \in B\}} \left| f(X^{(k+1)}) - f(X^{(k)}) \right| \\ &+ \sum_{\{k \in \{0,\dots,H-1\} | \text{pos } (X^{(k+1)} - X^{(k)}) \notin B\}} \left| f(X^{(k+1)}) - f(X^{(k)}) \right| \end{aligned} \tag{3.32}$$

For the second sum we want to use:

$$\left| f(X^{(k+1)}) - f(X^{(k)}) \right| \leq D_{\{1,\dots,n^2\}} \tag{3.33}$$

In the first sum we can make use of the specific column or row and use $D_{\mathcal{J}_j}$. This yields the following ansatz for $\lambda_{\mathcal{R}_j}$, which is the λ_k that belongs to the j-th row:

$$\lambda_{\mathcal{R}_j} = \alpha D_{\mathcal{R}_j} + \beta D_{\{1,\dots,n^2\}}. \tag{3.34}$$

To obtain the constant β we have to calculate the right side of the inequality:

$$\beta \geq \max_{\{\mathbf{x} \in \{0,1\}^{n^2} | A\mathbf{x} \neq \mathbf{b}\}} \frac{|n - I_{\max}|}{\sum_j |A\mathbf{x}_j - b_j|}. \tag{3.35}$$

If we have a row, where no element of $I_{\max}$ is present, there can be other ones placed there that share a column with a position in $I_{\max}$. If there is no one in that column, then $|A\mathbf{x}_j - b_j| \neq 0$ for the corresponding j. If there are ones then $\sum_j |A\mathbf{x}_j - b_j|$ also increases, since the ones are in places, where they share a column with a position in $I_{\max}$. This yields to the inequality:

$$\beta \geq \max_{\{\mathbf{x} \in \{0,1\}^{n^2} | A\mathbf{x} \neq \mathbf{b}\}} \frac{|n - I_{\max}|}{2 \cdot |n - I_{\max}|} = \frac{1}{2}. \tag{3.36}$$

To get an estimate for α we now consider a row that contains an element of $I_{\max}$. In the case, that there is only a single one we do not have to delete other ones. Every additional one increases $|A\mathbf{x}_j - b_j|$ also by one. Therefore:

$$\alpha \geq \max_{\{j, \mathbf{x} \in \{0,1\}^{n^2} | (A\mathbf{x})_j \neq b_j \quad \mathcal{J}_j \cap \text{vec}(I_{\max}) \neq \varnothing\}} \frac{|\text{vec}(B \setminus I_{\max}) \cap \mathcal{J}_j|}{\left| (A\mathbf{x})_j - b_j \right|} = 1. \tag{3.37}$$

3.10.2 Proofs of Lemma (3.2) and Proposition (3.3)

Proof of Lemma (3.2)

Proof. We express the set of permutation matrices in terms of the coefficients $x_{i,j}$:

$$\mathbb{P}_n = \{X \in \mathbb{R}^{n \times n} | \forall i, j \in \{1, ..., n\}\ x_{i,j} \in \{0, 1\}$$

$$\forall j \in \{1,...,n\} \sum_{i=1}^{n} x_{i,j} = \sum_{i=1}^{n} x_{j,i} = 1\}$$

$$= \Big\{ \begin{pmatrix} 1-\sum_{i=2}^{n} x_{i,1} & 1-\sum_{i=2}^{n} x_{i,2} & \dots & 1-\sum_{i=2}^{n} x_{i,n} \\ x_{2,1} & x_{2,2} & ... & x_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n,1} & x_{n,2} & ... & x_{n,n} \end{pmatrix}$$

$$\in \mathbb{R}^{n\times n} | \forall i \in \{2,...,n\}, j \in \{1,...,n\}\ x_{i,j} \in \{0,1\}$$

$$\forall j \in \{1,...,n\} \sum_{i=1}^{n} x_{j,i} = 1 \wedge \sum_{i=2}^{n} x_{i,j} \leq 1 \Big\}$$

$$= \Big\{ \begin{pmatrix} 1-\sum_{i=2}^{n} x_{i,1} & 1-\sum_{i=2}^{n} x_{i,2} & \dots & 1-\sum_{i=2}^{n} x_{i,n} \\ 1-\sum_{i=2}^{n} x_{2,i} & x_{2,2} & ... & x_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ 1-\sum_{i=2}^{n} x_{n,i} & x_{n,2} & ... & x_{n,n} \end{pmatrix}$$

$$\in \mathbb{R}^{n\times n} | \forall i \in \{2,...,n\}, j \in \{1,...,n\}\ x_{i,j} \in \{0,1\}$$

$$\forall j \in \{1,...,n\} \sum_{i=1}^{n} x_{j,i} = 1 \wedge \sum_{i=2}^{n} x_{i,j} \leq 1 \Big\}$$

$$= \Big\{ \begin{pmatrix} 1-(n-1)+\sum_{i,j=2}^{n} x_{i,j} & 1-\sum_{i=2}^{n} x_{i,2} & \dots & 1-\sum_{i=2}^{n} x_{i,n} \\ 1-\sum_{i=2}^{n} x_{2,i} & x_{2,2} & ... & x_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ 1-\sum_{i=2}^{n} x_{n,i} & x_{n,2} & ... & x_{n,n} \end{pmatrix}$$

$$\in \mathbb{R}^{n\times n} | \forall i \in \{2,...,n\}, j \in \{2,...,n\}\ x_{i,j} \in \{0,1\}$$

$$\sum_{i,j=2}^{n} x_{i,j} \in \{n-2, n-1\}\ \forall j \in \{1,...,n\}$$

$$\sum_{i=2}^{n} x_{j,i} \leq 1 \wedge \sum_{i=2}^{n} x_{i,j} \leq 1 \Big\}$$

$$= \Big\{ \begin{pmatrix} 2-n+\sum_{i,j=2}^{n} x_{i,j} & 1-\sum_{i=2}^{n} x_{i,2} & \dots & 1-\sum_{i=2}^{n} x_{i,n} \\ 1-\sum_{i=2}^{n} x_{2,i} & x_{2,2} & ... & x_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ 1-\sum_{i=2}^{n} x_{n,i} & x_{n,2} & ... & x_{n,n} \end{pmatrix}$$

$$\in \mathbb{R}^{n\times n} | \forall i \in \{2,...,n\}, j \in \{2,...,n\}\ x_{i,j} \in \{0,1\}$$

$$\sum_{i,j=2}^{n} x_{i,j} \in \{n-2, n-1\}\ \forall j \in \{2,...,n\} \sum_{i=2}^{n} x_{j,i} \leq 1 \wedge \sum_{i=2}^{n} x_{i,j} \leq 1 \Big\}$$

$$= \Big\{ \begin{pmatrix} 2-n+\sum_{i,j=2}^{n} x_{i,j} & 1-\sum_{i=2}^{n} x_{i,2} & \dots & 1-\sum_{i=2}^{n} x_{i,n} \\ 1-\sum_{i=2}^{n} x_{2,i} & x_{2,2} & ... & x_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ 1-\sum_{i=2}^{n} x_{n,i} & x_{n,2} & ... & x_{n,n} \end{pmatrix}$$

$$\in \mathbb{R}^{n\times n} | \forall i \in \{2,...,n\}, j \in \{2,...,n\}\ x_{i,j} \in \{0,1\}$$

$$\sum_{i,j=2}^{n} x_{i,j} \in \{n-2, n-1\}\ \forall j,i,k \in \{2,...,n\} \quad i \neq k$$

$$x_{j,i}x_{j,k} = 0 \wedge x_{i,j}x_{k,j} = 0 \Big\}$$

□

Proof of Proposition (3.3)

We want to find sufficient lower bounds for λ_1^j and λ_2 in

$$\begin{aligned}&\min_{\mathbf{x}\in\{0,1\}^{(n-1)^2}} \quad \mathbf{x}^{\mathrm{T}}\tilde{W}\mathbf{x} + \tilde{c}^T\mathbf{x} + \\ &\sum_{j=1}^{2(n-1)} \lambda_1^j \left(\sum_{k\in\mathcal{J}_j} x_k\right)\left(\sum_{k\in\mathcal{J}_j} x_k - 1\right) \\ &+ \lambda_2 \left(\sum_{i=1}^{(n-1)^2} x_i - (n-1)\right)\left(\sum_{i=1}^{(n-1)^2} x_i - (n-2)\right),\end{aligned} \tag{3.38}$$

so that it coincides with the constrained optimization problem. We make the following ansatz for the λ_2 parameter:

$$\lambda_2 = \frac{1}{2} D_{\{1,\ldots,(n-1)^2\}}, \tag{3.39}$$

since the function

$$\left(\sum_{i=1}^{(n-1)^2} x_i - (n-1)\right)\left(\sum_{i=1}^{(n-1)^2} x_i - (n-2)\right) \tag{3.40}$$

increases faster than $2 \cdot H$,when H is the number of entries that need to be switched in order to have $n-1$- or $n-2$-many entries equal to 1.

This choice for λ_2 allows us to only investigate the case, where we have $n-1$ or n ones placed anywhere. Similar to the prior proof we ask, what the worst ratio between ones we have to insert anywhere and ones we have to insert or delete in a particular column is. It can be easily seen that in the worst case scenario all ones are in one column or row. Therefore the ratio is one half. This shows that we can choose:

$$\lambda_1^j = \frac{1}{2} D_{\mathcal{J}_j} + \frac{1}{2} D_{\{1,\ldots,(n-1)^2\}}. \tag{3.41}$$

□

3.11 Experiments on IBM Burlington

Several of our experiments are conducted on the IBM five qubit QC-er in Burlington [1], which is accessible per cloud. In contrast to D-Wave quantum annealers, the circuit-model is used for the design of this quantum computer. From a purely theoretical perspective there is some equivalence [10] between adiabatic quantum computing and quantum computing with the circuit model, though due to the technical challenges, significant differences remain in practice. One way

to execute a time-dependent Hamiltonian $H(t)$ (3.6) on the IBM machine is to approximate it with L constant Hamiltonians. The Hamiltonians, which are constant in time, can be decomposed and expressed with quantum gates via *trotterization* [197].

More specifically, choosing the value of the parameter L in the piecewise constant approximation of the Hamiltonian is crucial as we loose the (pseudo-)adiabacity for small L, but – as each gate has a certain chance of introducing an error and a long execution time of the circuit makes errors due to decoherence more probable – large L are also prone to fail.

Because of the above-mentioned reasons, we are able to perform the adiabatic evolution for only two qubits on the 5 qubit processor until now. Fortunately, using the trick from Sec. 3.5.3, we can optimise over the 2×2 permutation matrices. In our experiment, we performed 50 time steps. We choose the evolution time 0.1 and use Suzuki expansion of order 2, similar to Kraus *et al.* [125]. The matrix W and the vector c were generated randomly for every of the 20 iterations. Every circuit was executed 8192 times, which is the maximal amount. An exemplary result of an execution is summarized in the histogram in Fig. 3.8. Although we obtain slightly different distributions each time, the highest peak always coincides with the second column of the optimal permutation, which is $|01\rangle$ here.

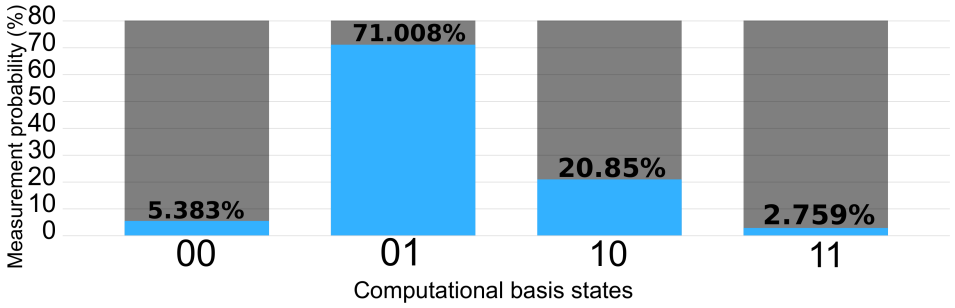


Figure 3.8: QGM execution histogram over 8192 runs using trotterization on a five qubit processor from IBM Quantum Experience. The states $|00\rangle$ and $|11\rangle$ are suppressed, because of the proposed regularization terms, *i.e.,* since these states violate that the sum of all elements in a column of a permutation matrix equals to 1.

The recent publication [53] about adiabatic quantum computing on machines from IBM Quantum Experience proposes catalyst Hamiltonian, which could improve the results in future and make it possible to succeed in higher dimensions.

3.12 Point Cloud Matching Example

To illustrate another application of the studied matching problems, we consider the registration of two 3D point clouds by finding correspondences of four pre-selected points in each scene as illustrated in Fig. 3.9. While this pre-selection is, of course, a very challenging part of the overall solution, our main goal is to illustrate the solution of a (low dimensional) matching problem via quantum computing – possibly as the solution to a subproblem in an iterative registration algorithm.

To set up the matching problem, we select four arbitrary points in one scene and use the

ground truth transformation between the two frames to identify the corresponding points in the other scene. We then set up a matching problem with costs as in (3.17) by simply using Euclidean distances between the points.

The histogram of energies obtained by all three quantum graph matching formulations over 500 anneals is shown in Fig. 3.10. The top row illustrates the overall histogram with strong peaks at low energies. As these peaks correspond to permutation matrices, we can conclude that all penalty terms were successful in strongly promoting permutations, with the inserted formulation yielding the best results. Zooming into the leftmost peak (illustrated in the bottom row of Fig. 3.10), however, reveals that none of the three formulations was successful in consistently predicting the global optimum among the permutation matrices. Considering the probabilities of less than 1% for the row-wise and baseline, and about 5.5% for the inserted formulations to predict the ground truth solution, one must conclude that all algorithms do not provide significantly better solutions than random guessing (which has a success probability of 4.17% for $n = 4$).

In summary, this experiment underlines the great difficulty current quantum hardware still has with problem instances of $n = 4$, *i.e.,* looking for the values of the best 16 qubits that are constrained to representing a permutation matrix. The performance of our three QUBO formulations on this problem can be seen in Fig. 3.10.

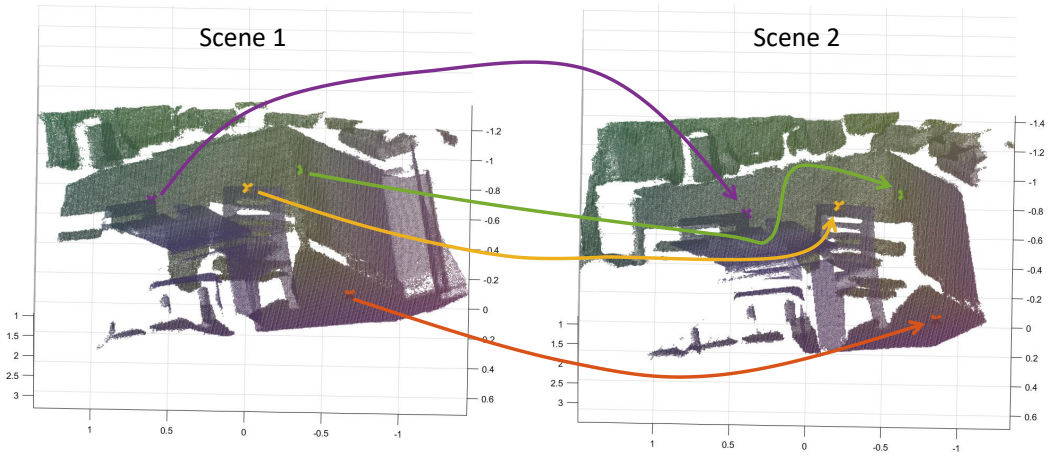


Figure 3.9: Illustrating the correct matching between four selected keypoints in two frames of the 7-Scenes 'Redkitchen' dataset, as provided at `https://3dmatch.cs.princeton.edu/`

3.13 Near-Isometric Matching

Fig. 3.11 shows an example of a matching problem between two shapes that are only approximately related by an isometry, *i.e.,* matching a wolf to a cat in this particular instance. Still modelling the problem as an isometric deformation using costs determined by (3.17) yields an instance, where the true matching still is the global optimum of the quadratic assignment problem, but where wrong permutations have considerably more similar energies than in the point cloud and isometric shape matching examples.

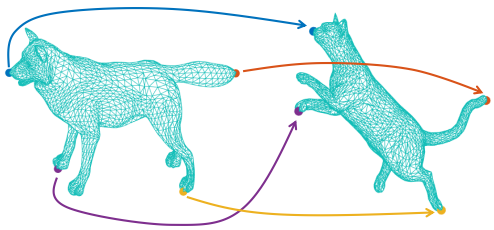


Figure 3.11: Illustrating a matching problem in which the shapes to be matched are only approximately related by an isometry.

The success probabilities for the *inserted*, the *baseline* and the *row-wise* QGM variants are 3.8%, 4% and 9.2%, respectively. As we can see in Fig. 3.12, the energy landscape has a wide range and the energy values corresponding to permutations are closer to each other compared to the case when the isometry assumption is strictly fulfilled. If one looks at the histograms with only permutations, only the *row-wise* QGM shows some trend towards the permutation with the lowest energy. Although its success probability is more than twice as good as random guessing, such a factor to random guessing would still not be sufficient to scale such an algorithm to large n.

3.14 Embedding to the Chimera Graph

The embeddings to the Chimera graph for different dimensions with the row-wise formulation can be seen in Fig. 3.13. While the number of logical qubits grows quadratically with n, the number of physical qubits required to embed those to the Chimera graph grows as n^4. For $n = 2$, the length of the longest chain is two physical qubits. For $n = 3$ and $n = 4$, the chain length does not exceed four and six, respectively.

3.15 Discussion of Negative Results

Since physical quantum computing is an emerging technology, reporting and discussing negative results on the early stage is of high relevance for the community. Insights of this section can help in choosing a promising direction for improvements and future research.

As discussed in section 3.6.2 we claim that the failure to provide the ground state with more probability than random guessing is due to the experimental errors in the coupling parameters. To back this up we provide the smallest and largest values of couplings and biases of the regularization and data term in table 3.2.

The couplings and the biases are scaled, so that they fit the feasible region of the annealer. $Q_{\text{reg}}, q_{\text{reg}}$ do yield constant energies for all permutations and therefore $Q_{\text{prob}}, q_{\text{prob}}$ contains the information, which permutation is optimal. The sum are the real, physical couplings

$$Q = Q_{\text{reg}} + Q_{\text{prob}}$$

Table 3.2: Illustrating the range of values arising from the penalty to constrain each formulation to permutation matrices ($Q_{\text{reg}\,*}$ and $q_{\text{reg}\,*}$) and from the actual problem costs ($Q_{\text{prob}\,*}$ and $q_{\text{prob}\,*}$). As we can see the constraints contribute more to the quadratic coupling matrix by a factor of around 6 for the inserted, 13 for the row-wise, and almost 55 for the baseline.

	Row-wise	Inserted	Baseline
Q_{max}	-0.887	-0.987	-0.973
Q_{min}	-1.037	-1.295	-1.009
q_{max}	-1.937	-8.884	-116.772
q_{min}	-2.277	-9.753	-120.772
$Q_{\text{reg max}}$	0.962	1.121	0.983
$Q_{\text{reg min}}$	-0.962	-0.705	-0.991
$q_{\text{reg max}}$	4.017	10.885	118.772
$q_{\text{reg min}}$	0.140	7.753	118.772

and biases

$$q = q_{\text{reg}} + q_{\text{prob}}.$$

One can see that most of the accessible range of the coupling parameters has to make sure that the output is a permutation.

There exists already ways to deal with these problems, which we want to try out in further experiments. One possibility would be to use extended J-range parameter [3]. The easiest way for this would be to use the virtual graph embedding instead of the default one (EmbeddingComposite). First attempts in this direction show that using the virtual graph embedding requires a lot of computation time. For one instance, where we got an error warning we had to invest 12% of our access time.

As reported in the main paper we first used a long annealing path with a break and at some point switched to using $20\mu s$. Although a longer annealing time can often be used to enhance the success probability, if we additionally look a the time it takes to perform the experiment until one gets the optimum with for example 99% certainty then according to [146] $20\mu s$ seems to be the better choice, if less than 512 qubits are used.

3.16 Beyond Quantum Computing

In addition to the numerical experiments using quantum computing, we also briefly tested the effect of our three reformulations on methods that are inspired by physical systems. In Fig. 3.14, we compare the success probability of our three formulations for simulated annealing using random instances of graph matching problems. As we can see, the row-wise and - even more so - the inserted formulations yield results clearly superior to the baseline method, indicating that our analysis might be of use beyond quantum computing. Due to the $\mathcal{NP}$-hard nature of the underlying problem, it is to be expected that the overall success probability still decreases exponentially with increasing n (for a fixed number of simulated annealing runs).

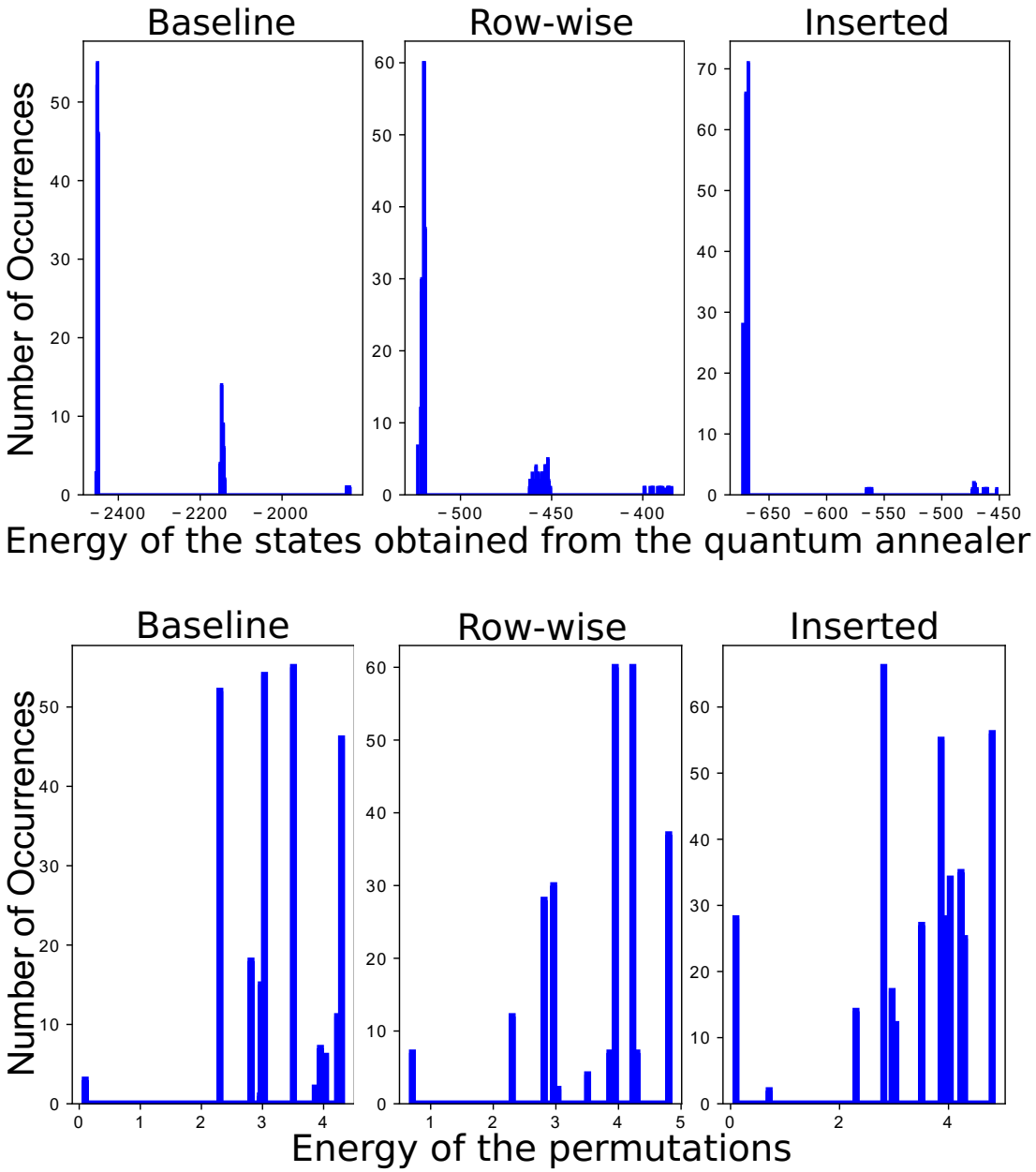


Figure 3.10: The histograms show the states obtained for the point matching problem in Fig. 3.9. The top row shows the overall histogram obtained over 500 anneals while the bottom row is a (rescaled) zoom into the leftmost peaks of the upper row, which corresponds to actual permutation matrices.

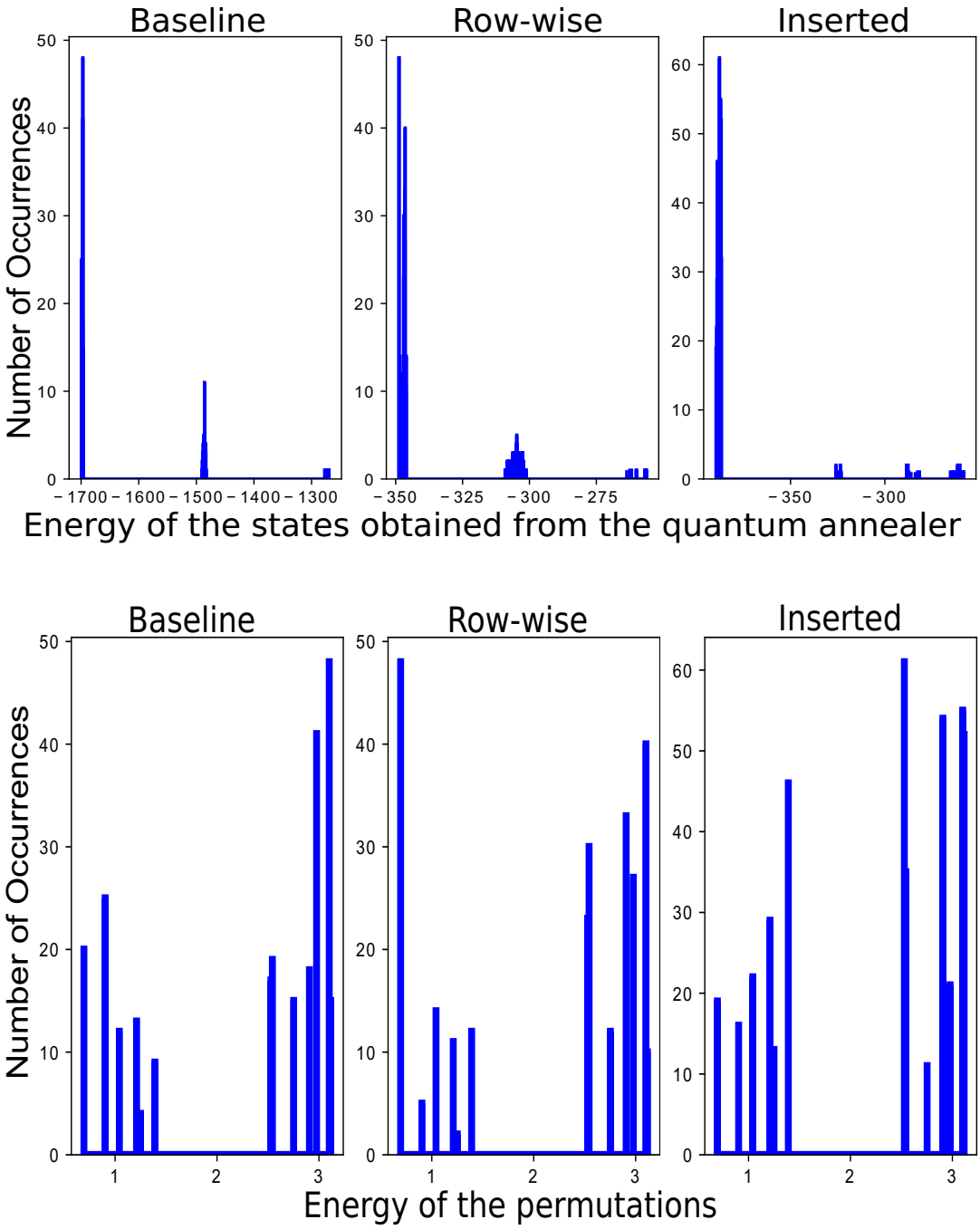


Figure 3.12: The histograms show the states obtained for the point matching problem in Fig. 3.11. The top row shows the overall histogram obtained over 500 anneals while the bottom row is a (rescaled) zoom into the leftmost peaks of the upper row, which corresponds to actual permutation matrices.

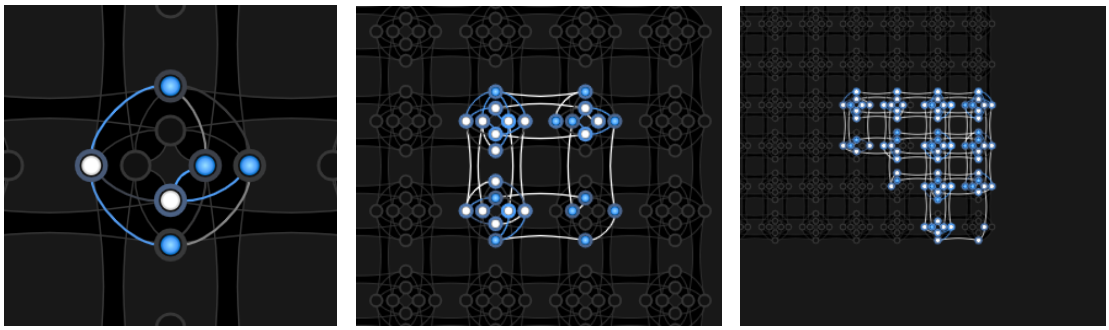


Figure 3.13: QGM minor embeddings to the chimera graph, for $n = 2$ (left), $n = 3$ (middle) and $n = 4$ (right). The number of logical qubits grows quadratically with n, and the number of physical qubits required to embed the logical qubits to the Chimera graph grows as n^4 on the current generation of D-Wave annealers. The white and blue circles denote measured values zero and one, respectively, and the lines between the qubits denote couplers. Grey lines connect the physical qubits with the same resulting values. A subset of those build chains and represent a single logical qubit. Blue lines connect different measured values. (Graphic generated using D-Wave leap problem inspector)

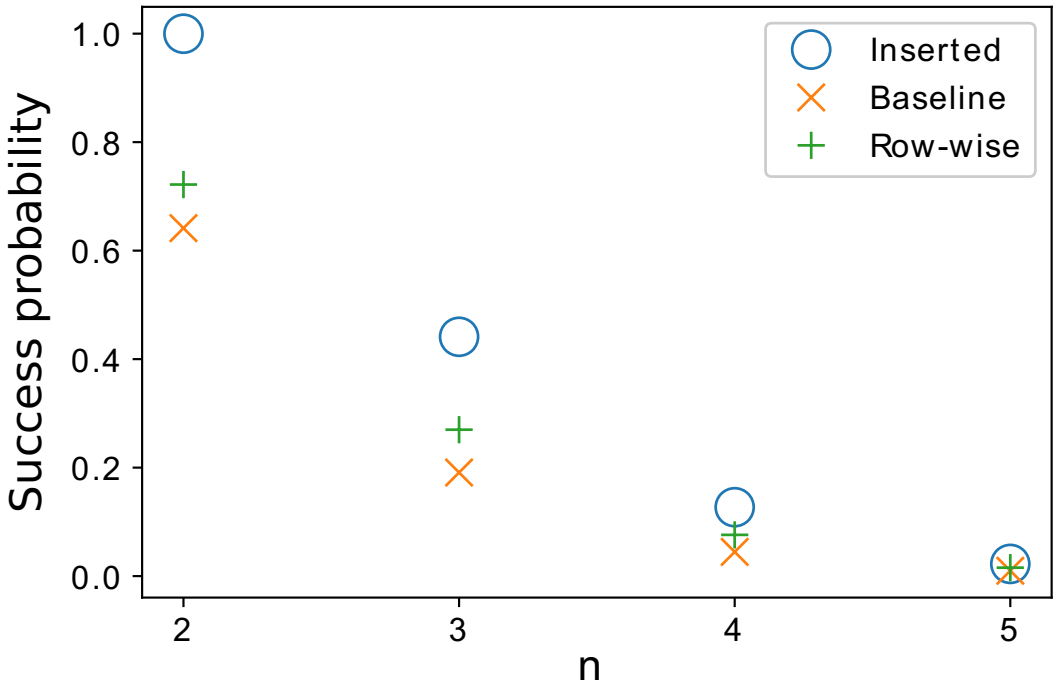


Figure 3.14: Percentage of the 5000 runs that found the optimum with simulated annealing averaged over 10 random problem instances.

Chapter 4

Q-Match: Iterative Shape Matching via Quantum Annealing

4.1 Contextualization

This chapter deals with the ICCV publication "Q-Match: Iterative Shape Matching via Quantum Annealing". This follow-up work of QGM uses an iterative algorithm and manages to come much closer to application relevant problem sizes. By iteratively optimizing which set of cycles should be applied to the current permutations to obtain the best energy improvement, one can avoid penalty terms. In this way, one can also mitigate problems with the limited precision in the couplings.

The idea to collaborate for this work came up because Zorah Lähner presented [128] at 3DV, where QGM was also presented, and because she joined the University of Siegen. Marcel Seelbach Benkner came up with the parameterization for the permutations used here and performed the experiments. Zorah Lähner made the work possible by providing expertise on the shape matching side. Michael Möller and Vladislav Golyanik significantly improved the design of the experiments and the scientific writing through close supervision. Christian Theobalt provided valuable discussion. Christof Wunderlich contributed by providing background knowledge about quantum annealing and notably improved the writing.

Abstract

Finding shape correspondences can be formulated as an $\mathcal{NP}$-hard quadratic assignment problem (QAP) that becomes infeasible for shapes with high sampling density. A promising research direction is to tackle such quadratic optimization problems over binary variables with *quantum annealing*, which allows for some problems a more efficient search in the solution space. Unfortunately, enforcing the linear equality constraints in QAPs via a penalty significantly limits the success probability of such methods on currently available quantum hardware. To address this limitation, this paper proposes Q-Match, *i.e.,* a new iterative quantum method for QAPs inspired by the α-expansion algorithm, which allows solving problems of an order of magnitude larger than current quantum methods. It implicitly enforces the QAP constraints by updating

the current estimates in a cyclic fashion. Further, Q-Match can be applied iteratively, on a subset of well-chosen correspondences, allowing us to scale to real-world problems. Using the latest quantum annealer, the D-Wave Advantage, we evaluate the proposed method on a subset of QAPLIB as well as on isometric shape matching problems from the FAUST dataset.

4.2 Introduction

Shape correspondence is at the heart of many computer vision and graphics applications, as knowing the relationship between points allows transferring information to new shapes. While this problem has been around for a long time [169], the setting with non-rigidly deformed shapes is still challenging [97]. One reason is that generic matching of n points in two shapes can be formulated as an $\mathcal{NP}$-hard Quadratic Assignment Problem (QAP)

$$\min_{X \in \mathbb{P}_n} \; E(X) := \mathbf{x}^{\mathrm{T}} W \mathbf{x}, \tag{4.1}$$

where $\mathbf{x} = \mathrm{vec}(X)$ is the n^2-dimensional vector corresponding to a matrix in the set of permutations $\mathbb{P} \subset \{0,1\}^{n \times n}$ and $W \in \mathbb{R}^{n^2 \times n^2}$ is an energy matrix describing how well certain pairwise properties are preserved between pairs of matches. For some choices of W and assumptions on the input shapes, (4.1) can become feasible by adding prior knowledge and exploiting geometric structures. For example, rigid shapes can be aligned through rotation and translation only — a problem with only six degrees of freedom. For general cases, QAPs remain challenging nonetheless.

In this work, we propose a new method for shape matching that employs *quantum annealing*, finds high quality solutions of QAPs with high probability, and is suitable for real-world problems. In recent years, quantum computers have made significant progress and machines such as the IBM Q System One (circuit model) and D-Wave Advantage System 1.1 (adiabatic, quantum annealer) became available for research purposes. The circuit-based and adiabatic models are polynomially equivalent to each other in theory [10]. However, the current implementations differ substantially and have different challenges to overcome. One advantage of adiabatic quantum computing is that it is less sensitive to noise and decoherence [60].

Unfortunately, adiabatic quantum computers (AQCer) can a-priori only solve unconstrained problems and thus cannot directly solve QAPs (4.1) which are constrained to permutation matrices. To overcome this limitation, a recent method [237] adds a penalty term enforcing the solution to be a permutation. However, this reduces the solvable problem sizes to very small $n \leq 4$ and, in practice, the probability of finding the globally optimal solution drops to random guessing. In contrast, we propose a new formulation for AQCing that is guaranteed to result in permutation matrices in problem sizes in the order of magnitude of available (fully connected) qubits. See Fig. 5.1 for the method overview and Table 4.1 for a list of acronyms we use. Our method is iterative, and we solve a series of quadratic unconstrained binary optimization (QUBO) problems on D-Wave which were prepared on a classical CPU. In summary, the **contributions** of this paper are as follows:

- Q-Match, *i.e.,* a new scalable quantum approach for searching correspondences between two shapes related by a non-rigid transformation (isometry or near-isometry).
- An iterative problem formulation through a variant of α-expansion which avoids explicit linear constraints for permutations and can be applied to large problem instances.

acronym	meaning
AQCer/ing	adiabatic quantum computer/ing
QPU	quantum processing unit
QAP	quadratic assignment problem
QUBO	quadratic unconstrained binary optimization

Table 4.1: Commonly used acronyms and their meaning.

We evaluate our method on a real AQCer, D-Wave Advantage system 1.1, and obtain improved results compared to the previous quantum method [237]. Moreover, Q-Match *performs on par with the classical matching approach [205].* This paper does not assume prior knowledge of quantum computing from readers. Secs. 4.4.1 and 4.4.2 review preliminaries required to understand, implement and apply the proposed Q-Match approach. The source code is available at `https://4dqv.mpi-inf.mpg.de/QMATCH/`.

4.3 Related Work

This section reviews prior work on the intersection of quantum computing, computer vision and shape matching. For the foundations of AQCing, see Sec. 4.4.1 and [139, 101].

AQCing for Computer Vision: Fueled by the progress in quantum hardware accessible for a broad research community [74, 133], a growing interest is developing to apply quantum computing or its concepts to computer vision. Ways for representing, retrieving and processing images on a quantum computer have been extensively investigated in the theory literature [263, 55, 248, 272]. Methods for image recognition and classification are among the first low-level techniques evaluated on a real AQCer [193, 45, 196, 58, 163]. O'Malley *et al.* [206] learn facial features and reproduce image collections of human faces with the help of AQCing. Cavallaro *et al.* [58] classify multi-spectral images with an ensemble of quantum support vector machines [268]. To account for the limited connectivity of the physical qubit graph of D-Wave 2000Q, they split the training set into multiple disjoint subsets and train the classifier on each of them independently. Li and Ghosh [163] eliminate false positives in multi-object detection on D-Wave 2X using the QUBO formulation from [228]. It provides state-of-the-art accuracy and the implementation with quantum annealing is faster than greedy and tabu search.

Solving image matching problems with a quantum annealer was first theoretically studied in [194]. Practically applicable quantum algorithms for the absolute orientation problem and point set alignment have been introduced in [111]. The work shows that representing rotation matrices using a linear basis allows mapping both problems to QUBOs. A new point set alignment method for gate model quantum computers has been recently derived in [198], inspired by the earlier kernel correlation approach [258]. QGM [237] is the first implementation of graph matching for small problem instances as a single QUBO with a penalty approach, whereas QuantumSync [40] is the first method for permutation synchronization developed for and tested on an AQCer (Advantage system 1.1). In [237], the formulation of permutation matrix constraints with a linear term leads to low probabilities of measuring globally optimal solutions in a single annealing cycle on a modern AQCer. In contrast to [237, 40], we avoid explicit constraints ensuring valid permutations and use a series of QUBOs leading successive improvements in the energy. Consequently, we can

align significantly larger graphs and shapes.

Shape Correspondence: Finding point-wise matches between meshes is an actively studied problem in vision and graphics. One common way to establish matches between shapes related by isometry is to compare shape functions or signatures. Multiple methods of this category analyze spectra of the Laplace-Beltrami operator on the shape surfaces which are invariant under isometric deformations [249, 22, 205, 232, 122]. Functional maps [205, 122] interpret the problem as the alignment of functions in the pre-defined basis (*e.g.,* eigenfunctions of the Laplace-Beltrami operator). Post-processing is required to extract point-wise matches. Among the shape descriptors, wave kernel signature (WKS) [22] is inspired by quantum physics. It relies on solving the Schrödinger equation for the dynamics (dissipation) of quantum-mechanical particles on the shape surface. WKS can resolve fine details and is robust to moderate non-isometric perturbations. Recently, convolution operators were generalized to non-Euclidean structures, and the availability of large-scale shape collections enabled supervised learning of dense shape correspondences [177, 189, 164].

The point-wise correspondence search between two shapes can be formulated as a linear (LAP) or quadratic assignment problem (QAP) over the space of permutations, with the matching costs computed relying on feature descriptors [249, 22, 232]. The solution space of both problems is exponential in the number of points. For LAP, there exists a fast auction algorithm with polynomial runtime [37]. Per construction, LAP formulations for matching are sensitive to surface noise and do not explicitly consider spatial point relations, which often leads to local minima (and, consequently, inaccurate solutions). QAPs [153, 158] add quadratic costs for matching point pairs and regard point neighborhoods; the solutions are spatially smoother. The downside is their $\mathcal{NP}$-hardness, which makes finding global optima for large inputs unfeasible. Multiple policies to solve QAPs efficiently have been proposed in the literature, such as branch-and-bound [227, 120], spectral analysis [162], alternating direction method of multipliers [159], entropic regularization of the energy landscape [245] and simulated annealing [128]. Convex relaxations are among the most thoroughly investigated policies for QAPs [277, 15, 217, 104, 140, 33]. These methods either have prohibitive worst-case runtime complexity (branch-and-bound), rely on heuristic algorithmic choices (*e.g.,* entropic regularization and simulated annealing), or do not guarantee global optima. In contrast, we address QAPs with a new AQCing metaheuristic to find high quality solutions with a high probability.

Holzschuh *et al.* [128] proposed a non-quantum algorithm closely related to ours. It can be seen a special case of our setup that only applies one 2-cycle per iteration. This means a considerably smaller step size compared to Q-Match and, hence, more iterations until convergence and higher sensitivity to local optima. Several heuristics to gain robustness and speed are further used in [128]. First, the algorithm is based on simulated annealing and accepts worse states with a certain probability to step over local optima. In contrast, our Q-Match explores larger solution spaces simultaneously and converges in fewer iterations. Second, their hierarchical optimization scheme adds new matches based on geodesic embeddings whereas we can refine any given initialization.

4.4 Background

4.4.1 Adiabatic Quantum Annealing (AQC)ing

The seminal work of Kadowaki and Nishimori [139] introduced quantum annealing. The authors argued that a quantum computer could be built based on the principle of finding the ground state of the Ising model under quantum fluctuations (causing state transitions) induced by a transverse magnetic field. The theoretical foundation of such a computational machine is grounded on the adiabatic theorem [44]. In follow-up work, Farhi *et al.* [101] tested the quantum annealing algorithm simulated on a classical computer on hard instances of an $\mathcal{NP}$-complete problem. Although it is *not* believed that quantum annealing can solve $\mathcal{NP}$-complete problems in polynomial time, for rugged energy landscapes with high and narrow spikes quantum annealing can be faster than simulated annealing [81, 86] and for some instances D-Wave scales better than the classical path integral Monte Carlo method [147].

AQCing can solve *quadratic unconstrained binary optimization* (QUBO) problems which read

$$\min_{s\in\{-1,1\}^n} s^\top Js + b^\top s, \tag{4.2}$$

where s is a binary vector of unknowns, J is a symmetric matrix of inter-qubit couplings, and b contains qubit biases. To change the binary variable to $x \in \{0,1\}^n$ one simply needs to insert $s_i = 2x_i - 1$ everywhere in (4.2).

Quantum computers operate with *qubits*, *i.e.,* quantum mechanical systems which can be modeled with normalized vectors $|\phi\rangle$ in a Hilbert space $\mathbb{H}$. One writes $|\phi\rangle = \alpha|0\rangle + \beta|1\rangle$ with $\alpha, \beta \in \mathbb{C}$ and $|\alpha|^2 + |\beta|^2 = 1$, where $|0\rangle, |1\rangle$ denote an orthonormal basis. In contrast to classical bits, which are either in state 0 or 1, quantum mechanical systems can be in a superposition $\alpha|0\rangle + \beta|1\rangle$, where after measurement in the so-called computational basis ($\{|0\rangle, |1\rangle\}$) one obtains $|0\rangle$ with probability $|\alpha|^2$ and $|1\rangle$ with probability $|\beta|^2$. A composite system of, *e.g.,*, two qubits can be described with vectors from the tensor product of the individual Hilbert spaces $|\psi\rangle \in \mathbb{H}\otimes\mathbb{H}$. If two qubits can be described independently from each other, we have $|\phi\rangle \in \mathbb{H}$ and $|\eta\rangle \in \mathbb{H}$, and the two-qubit state $|\psi\rangle$ is described by the product state $|\psi\rangle = |\phi\rangle \otimes |\eta\rangle$. Since the state space of n qubits is 2^n dimensional, a classical computer would have problems even storing all the coefficients for $n = 50$ [197]. A second ingredient required for quantum computing, in addition to the superposition of qubit states, is the interference between different possible computational paths in the 2^n-dimensional state space caused by interactions between qubits. Thus, during the course of a quantum algorithm *entangled* (non-classically correlated) states such as $\frac{1}{\sqrt{2}}(|0\rangle\otimes|0\rangle + |1\rangle\otimes|1\rangle)$ are created that cannot be decomposed in the above form. Entanglement between quantum states is often considered a necessary resource for quantum computing [218]. At the end of a quantum algorithm the coefficients α and β for each qubit are determined in a suitable measurement leaving the (adiabatic or circuit-model) quantum computer in a classical (non-entangled) state.

Another central notion in quantum mechanics are linear *Hamilton operators*, acting on elements of $\mathbb{H}$, which are used to describe the static and dynamic properties of a quantum system with the Schrödinger equation. The eigenstates of a (time-independent) Hamiltonian are the stationary energy states of a quantum system. The eigenvalues give the corresponding possible values of the system's energy. The key idea for solving problems like (4.2) is to prepare a quantum system in a

known state of lowest energy of a Hamiltonian H_I, most commonly a product state of the form,

$$|\psi(t=0)\rangle = \bigotimes_{i=1}^{n} \frac{1}{\sqrt{2}} \left(|0\rangle + |1\rangle\right), \tag{4.3}$$

with "$\bigotimes$" denoting the Kronecker product over n subsystems, and each qubit is prepared in a superposition. Next, one constructs a *problem Hamiltonian* H_P in such a way that the lowest energy state of H_P is a solution of (4.2). Finally, one slowly switches from the initial Hamiltonian H_I to the final Hamiltonian H_P with, *e.g.,* a linear combination

$$H(t) = \left(1 - \frac{t}{\tau}\right) H_I + \frac{t}{\tau} H_P. \tag{4.4}$$

If the transition in (4.4) is sufficiently slow (*i.e.,* if τ is sufficiently large), and if the state of lowest energy (ground state) of $H(t)$ remains unique, the adiabatic theorem [44] guarantees that the quantum system will remain in the ground state for all t. Thus, the solution to (4.2) can simply be *measured* after the time evolution starting in an eigenstate of H_I (that is typically easy to prepare) and ending in an eigenstate of H_P. The theoretical requirements for adiabatic evolution can be made precise in terms of the size of the *spectral gap* of $H(t)$, *i.e.,* the difference between the smallest and second smallest eigenvalues of $H(t)$, during the time evolution. Building a system to actually prepare a quantum system and evolve it in an adiabatic way in practice, however, remains highly challenging. A spectral gap that is too small will lead to excitation of higher energy states during the annealing process, thus preventing the system from ending up in the desired ground state of H_P. Next, we discuss some details regarding such practical realizations.

4.4.2 Algorithm Design and Programming the D-Wave Quantum Annealer

Every AQCing algorithm includes six steps, *i.e.,* QUBO preparation, minor embedding, quantum annealing (sampling), unembedding, bitstring selection (in the case of multiple annealings) and solution interpretation.

QUBO preparation: Note that many computer vision problems are naturally formulated in forms other than a QUBO. A lot of research thus focuses on the QUBO preparation, *i.e.,* how to formulate a target problem as a QUBO mappable to an AQCer (Sec. 5.5 is devoted entirely to the QUBO preparation for Q-Match). Note that this step is performed entirely on the CPU. In QUBOs, each binary variable is interpreted as one *logical* qubit in the quantum context. Thus, the biases b and couplings between the qubits J define a graph of logical problem qubits.

Minor embedding is the mapping of logical qubits to the grid of physical hardware qubits of the AQCer. Due to the limited connectivity between the physical qubits, each logical qubit often requires several physical qubits in the minor embedding. Physical qubits realizing a single logical qubit—and having as similar quantum states during annealing as possible—build a *chain.* To find a minor embedding for two arbitrary graphs is an $\mathcal{NP}$-hard optimization problem on its own [52]. However the problem becomes easier since the one graph is fixed by the hardware and heuristic methods such as [52] can be used. The core criteria for an optimal minor embedding is

jointly minimizing the chain length and the total number of required physical qubits.

Quantum annealing is initiated, once the minor embedding is complete. It corresponds to the evaluation of the system with respect to a time-dependent Hamiltonian such as (4.4) during which the quantum system ideally remains in the ground state during the entire annealing, see Sec. 4.4.1. In practice, due to such factors as 1) the expected qubits lifetime (*e.g.,* influenced by interaction with the environment, which cannot be entirely prevented), and 2) a small spectral gap of $H(t)$, multiple anneals are needed, each of which leads to the global optimum with some probability <1. After each annealing, an **unembedding** algorithm assigns measured values to the logical qubits.

The final solution over multiple annealings is chosen in the **bitstring selection** step based on the occurrence frequency or the resulting energies. Finally, the solution is **interpreted** in the context of the original problem and returned to the user. In this regard, the interpretation involves transformation of the measured bitstring to the initial data modality (*e.g.,* permutation matrices).

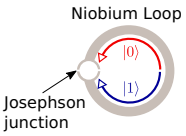


Figure 4.1: Superconducting flux qubit.

D-Wave programming is done in Python, using the *Leap 2* programming interface and remote access tools [74]. The D-Wave machine uses superconducting flux qubits [182]. The qubits and the couplers are realized with loops of niobium, which have to be cooled to temperatures below 17mK. The current, that flows clockwise or anti-clockwise through the niobium loop can be modeled as a qubit, see Fig. 4.1. Thus, it is possible to have a superposition between a current flowing clockwise and anticlockwise as long as no measurement is performed. The specified couplings J and biases b in (4.2) determine the time evolution of the magnetic fluxes during quantum annealing.

4.4.3 α-Expansion

The α-expansion algorithm was introduced in [46] to solve labeling optimization problems (*e.g.*, for semantic segmentation or disparity estimation). A notable extension is [161], where the method is generalized to allow for parallel computation amongst other things. In each iteration, a vector of binary variables α that correspond to two proposed values for the labels is computed so that the decision between the proposals improves the energy. [46] finds the optimal expansion through a graph cut such that each pixel in an image is assigned to the first proposed label if $\alpha_i = 1$ or the second proposed label if $\alpha = 0$.

Apart from this application, α-expansion works particularly well in case the subproblem, *i.e.,* the optimization for α, is submodular. Then the subproblem can be solved efficiently using graph cut techniques.

4.4.4 Cyclic Permutations

We define the set of permutation matrices as $\mathbb{P}_n = \{X \in \{0,1\}^{n\times n} \mid \sum_i X_{ij} = 1,\ \sum_j X_{ij} = 1\ \forall i,j\}$. A permutation matrix X is called a *k-cycle*, if there exist k many disjoint indices $i_1,\dots,i_k$ such that $X_{i_1 i_k} = 1$ and $X_{i_{j+1} i_j} = 1$ for all $j = 1,\dots,k-1$, and $X_{ll} = 1$ for all $l \notin \{i_1,\dots,i_k\}$, in which case $X = (i_1\ i_2\ \dots\ i_k)$ is a common notation. Two permutations X and X' are called *disjoint* if $X_{ii} \neq 1$ implies $X'_{ii} = 1$ and $X'_{ii} \neq 1$ implies $X_{ii} = 1$. It is easy to show that disjoint permutations commute. Furthermore, it holds that any X can be written as the product of 2-cycles c_i, *i.e.,* $X = \prod_{i=0}^{N} c_i$. Finally, we use the notation $X^1 = X$ and $X^0 = I$ where I is the identity for any matrix X.

4.5 Method

The main difficulty of previous quantum computing methods such as [237] for solving (4.1) is handling the linear equality constraints arising from the optimization over permutation matrices. Instead of enforcing them via a penalty, the core idea of Q-Match is to minimize (4.1) iteratively: We propose to use a variant of the α-expansion algorithm that chooses k-many disjoint candidate cycles c_i and uses quantum computing to solve the (still $\mathcal{NP}$-hard but smaller and unconstrained) subproblem of deciding whether to apply c_i or not (identified with $\alpha_i = 1$ and $\alpha_i = 0$, respectively). In the following, we detail the idea of the cyclic α-expansion as well as our proposed Q-Match algorithm.

4.5.1 Cyclic α-Expansion

Let $C = \{c_1, ..., c_m\}$ be a set of m disjoint cycles. We consider the following optimization problem for an initial permutation P_0:

$$\underset{\{P\in\mathbb{P}_n \mid \exists \alpha\in\{0,1\}^m:\ P=\left(\prod_i c_i^{\alpha_i}\right)P_0\}}{\arg\min} E(P), \tag{4.5}$$

where E is defined in (4.1) and α is a binary vector parametrizing P. As noted in Sec. 4.4.4, the order in the product of all $c_i^{\alpha_i}$ does not matter as disjoint permutations commute. (4.5) would be equivalent to (4.1) if C was not fixed with disjoint cycles but could also be optimized.

Problem complexity: The optimization problem in (4.5) makes a binary decision for each cycle whether it should be applied. In this section, we show that (4.5) is an optimization problem with complexity dependent on the number of cycles and not n. We can convert the multiplication of cycles from (4.5) into the following linear combination

$$P(\alpha) = P_0 + \sum_{i=1}^{m} \alpha_i (c_i - I) P_0, \tag{4.6}$$

where I is the identity. See the supplement for a proof. Next, we use this representation to show that (4.5) is a problem in the form of (4.2) with size m. Let P, Q be two permutations and $E(Q,R) = \mathrm{vec}(Q)^T W \mathrm{vec}(R)$ which is linear in each component. We write $C_i = (c_i - I)P_0$ to

Algorithm 2 Q-Match

Input: Shapes M, N
Output: P

```
Initialize P (Sec. 4.5.2)
repeat
    Calculate the k worst vertices I_M, I_N (4.10)
    Construct sub-matrix of worst matches W_s (S. 4.5.2)
    repeat
        Choose random set of new 2-cycles
        Calculate W̃ (Sec. 4.5.2 and (7))
        Solve the QUBO (4.8) with AQCing
        Update permutation according to (4.6)
    until Every 2-cycle occurred in one set
    Apply permutation to worst vertices in P
until P does not change anymore
return P
```

simplify the notation. Thus, to solve (4.5), we need to minimize

$$\begin{aligned}
&E(P_0 + \sum_i \alpha_i C_i, P_0 + \sum_j \alpha_j C_j) \\
&= E(P_0, P_0 + \sum_j \alpha_j C_j) + \sum_i \alpha_i E(C_i, P_0 + \sum_j \alpha_j C_j) \\
&= E(P_0, P_0) + \sum_j \alpha_j E(P_0, C_j) + \sum_i \alpha_i E(C_i, P_0) \\
&+ \sum_i \sum_j \alpha_j \alpha_i E(C_i, C_j) = E(P_0, P_0) + \alpha^T \tilde{W} \alpha,
\end{aligned}$$

with

$$\tilde{W}_{ij} = \begin{cases} E(C_i, C_j) & \text{if } i \neq j, \\ E(C_i, C_i) + E(C_i, P_0) + E(P_0, C_j) & \text{otherwise.} \end{cases} \tag{4.7}$$

As $E(P_0, P_0)$ is constant w.r.t. α, we are left with

$$\min_{\alpha \in \{0,1\}^m} \alpha^\top \tilde{W} \alpha, \tag{4.8}$$

which can be solved directly by AQCers. The interpretation of the optimal α is a binary indicator for whether to apply the corresponding cycle or not. With the basic methodology of (4.8) being established, we proceed iteratively: Given a current estimate P_{i-1} minimizing (4.1), we choose a set of disjoint cycles $C = \{c_1, c_2, \ldots, c_m\}$, optimize (4.8), and set $P_i = \left(\prod_j c_j^{\alpha_j}\right) P_{i-1}$.

4.5.2 Q-Match

Cyclic α-expansion can optimize any QAP up to a problem size that fits within the AQC, *e.g.,* by iterating over random sets of cycles. To solve larger problems of (isometric) shape correspondence

between two 3D shapes M and N, we propose *Q-Match*. If both shapes are discretized with the same mesh of n vertices, W has the following form:

$$W_{i\cdot n+k,j\cdot n+l} = |d_M^g(i,j) - d_N^g(k,l)| \tag{4.9}$$

where i,j are vertices on M, k,l are vertices on N, and $d^g(a,b)$ is the geodesic distance between two points on the same shape. Therefore, for two matches $(i,k),(j,l)$ $W_{i\cdot n+k,j\cdot n+l}$ describes how well the geodesic distance is preserved between the elements of this pair. The optimal solution is the correspondence for which the distance between all pairs of points is preserved the best, for isometries the optimal energy of (4.1) is zero. Since the geodesic distances have geometric meaning, this propagates into the QAP, and Q-Match leverages this to choose better sets of cycles. In Q-Match, the set of tested cycles is based on the contribution of each vertex to (4.1) (Sec. 4.5.2), we can solve subproblems iteratively (Sec. 4.5.2), and the initial permutation is descriptor-based (Sec. 4.5.2). An overview of the entire Q-Match algorithm can be found in Alg. 2.

Choosing Cycles

One option to select C is by randomly drawing disjoint cycles, which might, however, take many iterations to reach the optimum. Instead we make use of the fact that the isometric shape matching problem creates QAPs in which the entries are highly correlated.

Worst Vertices: For a point with index v in M and a permutation P, we can quantify the influence of v on (4.1) via

$$I(v) = \sum_{w\in M} W_{v\cdot n+P(v),w\cdot n+P(w)}, \tag{4.10}$$

where $P(v)$ denotes the index v is mapped to. For vertices on N we proceed equivalently. If $I(v)$ is high, this is an indicator for $P(v)$ being inconsistent with the majority of other matches in P. Therefore, we collect the m vertices with highest values for I on each shape in the two sets I_M and I_N. The potential improvement of (4.1) is maximized by this choice. See Sec. 4.6.3 for an analysis of the size of m.

2-Cycles: Subsequently, we choose a set of k random but disjoint 2-cycles from I_M, I_N. While the limitation to 2-cycles might sound restrictive, the following lemma (proven in the appendix and in many abstract algebra textbooks [235]) highlights the expressive power of disjoint 2-cycles:

Lemma 4.1. *Every permutation P can be written as $P = QR$, where Q and R are products of disjoint 2-cycles.*

Subproblems

3D meshes normally consist of more than thousand vertices, this means $W \in \mathbb{R}^{n^2\times n^2}$ cannot be precomputed and stored in memory. Therefore, in each iteration all needed entries of W to compute $\tilde{W}$ for (4.8) have to be computed from scratch. As can be seen in (7) several evaluations of the cost function E are needed for $\tilde{W}$ which makes this operation expensive.

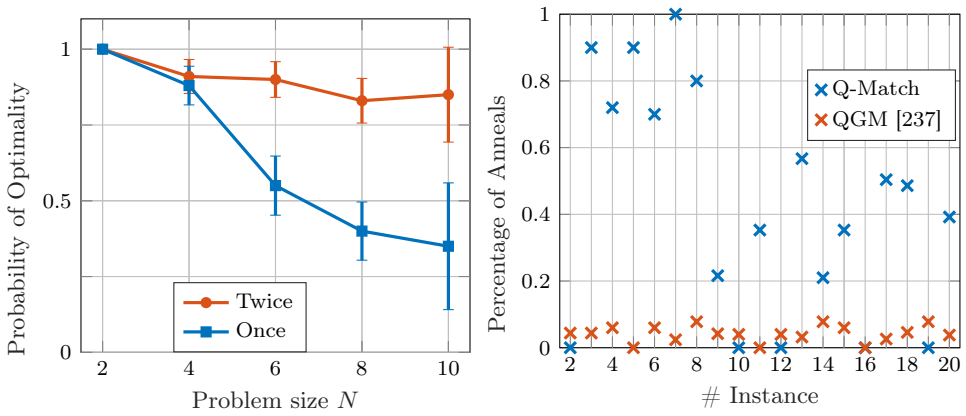


Figure 4.2: Experiments on how often Q-Match retrieves the optimum for (4.1). (Left) Fraction of random instances with increasing problem size for which our iterative method leads to the optimum with brute-force calculation. (Right) Success rate of Q-Match and QGM [237] on 20 random problems. We count (QGM) or lower bound (ours) the percentage of anneals that reach the optimum.

To be more efficient, we split the computation into the calculation of W_s, a $k^2 \times k^2$ reduction of W based on the k worst vertices in I_M, I_N. This is the most expensive operation in our algorithm (see Fig. 4.5). For a set of 2-cycles on I_M, I_N, computing $\tilde{W}$ is now efficient. To reduce the number of times we have to compute W_s, instead of just choosing one set of 2-cycles on each I_M, I_N, we evaluate several sets. In practice, we sample sets of 2-cycles on I_M, I_N until every possible 2-cycle has been chosen at least once. A detailed description how to compute W_s and $\tilde{W}_s$ and a runtime analysis can be found in the supplementary material.

Initialization

The only thing left is choosing P_0 for the first iteration. Instead of a random initialization, we calculate a descriptor-based similarity $S \in \mathbb{R}^{N \times N}$ between all vertices of M and N, equivalent to the strategy of [265]. S_{mn} contains the inner product of the normalized HKS [249] and SHOT [255] descriptors of vertices $m \in M, n \in N$, and we solve a linear assignment problem on S to get the first permutation.

4.6 Experiments

We evaluate accuracy, convergence, and runtime of our method on random QAP instances (Sec. 4.6.1), FAUST (Sec. 4.6.2) and QAPLIB (Sec. 4.6.5). Since quantum computing time is still extremely expensive, we only evaluate on subsets of FAUST and QAPLIB, but our results show how promising this technology can be in the future. All experiments were done on Intel i5 8265U CPU with 16GB RAM using Python 3.8 and D-Wave Advantage system 1.1 accessed by Leap 2. Note that one can use classical QUBO solvers for the subproblems. In the appendix, we present experiments for this with a simulated annealing sampler.

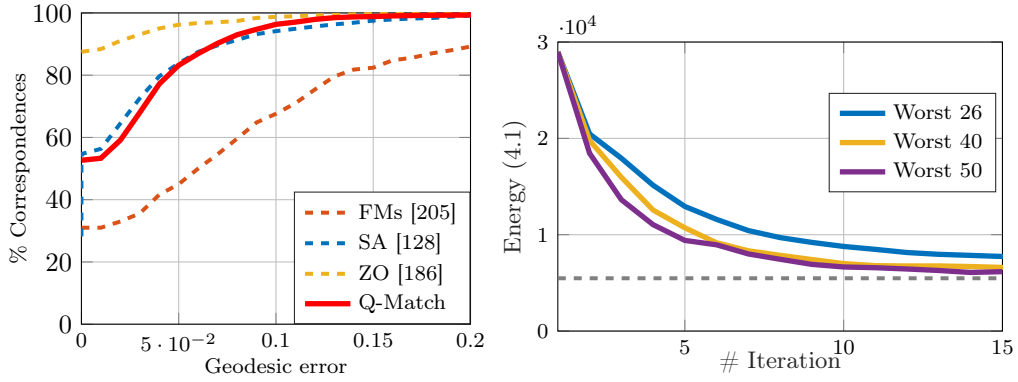


Figure 4.3: Evaluation of cumulative error [144] (left) and convergence (right) on FAUST. (Left) We compare against Simulated Annealing [128] without post-processing and Functional Maps [205]. Dashed lines indicate non-quantum approaches. The results have symmetry-flipped solutions removed, these have a comparable final energy for all methods but are not recognized as correct in the evaluation. (Right) We show convergence of the energy over 30 iterations. The larger the set of worst vertices, the faster our method converges. The dashed grey line shows the optimal energy.

4.6.1 Comparison to Penalty Based Implementation

We compare Q-Match to the inserted method in [237] for problems of the form (4.9) with distances $d_{i,j}$ drawn uniformly at random from $[0, 1]$, and W being constructed using a randomly drawn true permutation P. We compare the success probability of both methods on 20 random instances, see Fig. 4.2-(right). Q-Match finds the optimum in all but 4 cases. For a fair comparison, we restrict the number of anneals per iteration in our method to 10 and use 500 anneals for the inserted method. Additionally, the success probability of the iterative method is bounded by the success probabilities of the individual steps multiplied with each other. In the case that multiple distinct outputs have the lowest energy, both results will be counted as success. For $n \in \{2, 4, 6, 8, 10\}$, we count how often the optimal permutation is reached, if each subproblem is solved to global optimality. For this we generated 20 random instances for $n = 10$ and 100 random instances for the other dimensions. The errors are calculated as a binomial proportion confidence interval that should contain the true probability in 95% of the cases. This experiment was performed first so that all 2-cycles occur only once and a second time where the whole process is carried out twice, see Fig. 4.2-(left).

Moreover, we study if the solution with the lowest value returned by the quantum annealer is globally optimal. For 500 runs and coupling matrices up to a size of 13 this was always the case. Therefore, the number of optima found by Q-Match is identical to the probabilities in Fig. 4.2-(left).

4.6.2 FAUST

We perform experiments on the registrations of the FAUST dataset [41]. We evaluate on 5 different pairs from one class within FAUST that were downsampled to $502f$ vertices. It is in theory possible to use our method on the original resolution and more pairs, but QPUs are not as

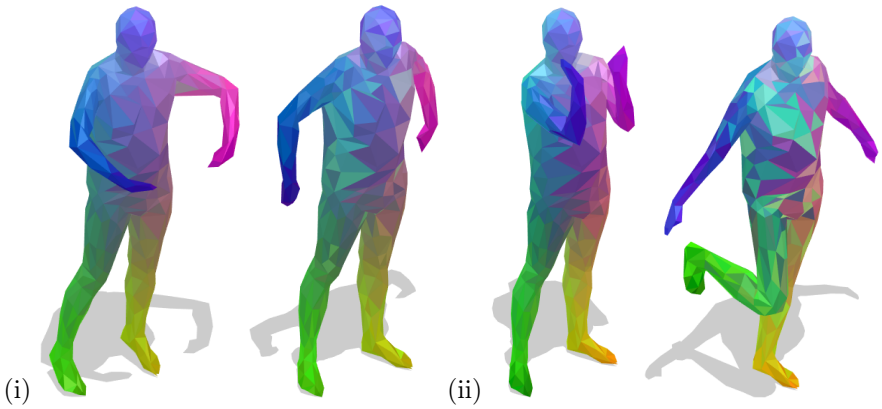


Figure 4.4: Example correspondences from the FAUST registrations. (i) Our results are overall correct with some outliers. (ii) A failure case. The solution partially swapped front and back. These kinds of solutions are near-isometries, have energies close to the optimum, and often occur in all methods that purely solve (4.1).

freely available as traditional hardware, and we focus our available computing time on a deeper analysis of our method. We apply Q-Match as described in Alg. 2 with 500 anneals, and compare to two non-quantum methods, Simulated Annealing [128] and Functional Maps [205]. [128] is applied without ZoomOut [186] such that it is close to a version of our algorithm where only one 2-cycle is chosen in each iteration. The functional maps implementation includes a Laplacian commutativity regularizer and retrieves the final point-to-point correspondence through an LAP [205]. While [237] is the conceptionally closest method to ours, it only produces matches for up to preselected 4 vertices. The quantitative evaluation can be seen in Fig. 4.3, and some qualitative examples of our method in Fig. 4.4. While we do not reach the same accuracy as the recent [128], we do outperform the classical but still popular functional maps. Since AQCers are still newly developed and have limited complexity in comparison to traditional computers, we see this as a successful application of quantum annealing for real-world problems.

4.6.3 Convergence

We evaluate how many iterations Q-Match needs to converge on FAUST. Fig. 4.3 shows the convergence when choosing $26, 40, 50$ worst vertices in each iteration. Since the margin between 40 and 50 is small, we chose to run all further experiments with the 40 worst matches. In each iteration every 2-cycle occurs once in a set C. Some exemplary matches obtained in this way can be seen in Fig. 4.4.

4.6.4 Runtime

Fig. 4.5 shows how the runtime of Q-Match scales with the problem size given by the number of worst vertices. The calculation of the submatrix W_s is the most expensive. Computing the matrix $\tilde{W}$ given W_s takes only ms, while the calculation of W_s is in the order of seconds. We did not optimize or parallelize this operation though. The QPU access time is the time the quantum annealer spends to solve the problem, and is mostly independent of the problem size.

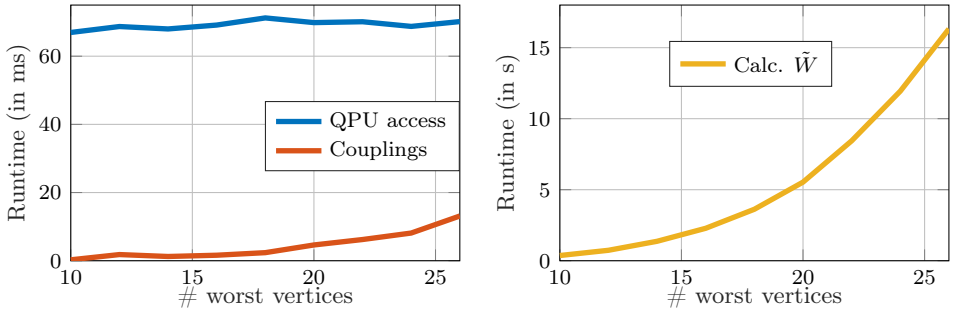


Figure 4.5: Influence of the problem size on the runtime. We increase the size of the set of worst vertices (see Sec. 4.5.2) and measure the runtime development of different parts of our pipeline. While the QPU access time is mostly independent of the problem size, calculating $\tilde{W}$ on the CPU grows drastically.

We report results for about 25 minutes of QPU time, which allows us to evaluate 10^4 QUBOs. One FAUST instance using 40 worst vertices and 500 anneals per QUBO requires about 2min of QPU time. For all experiments we used the default annealing path and the default annealing time of 20μs. The chain strength was chosen as 1.0001 times the largest absolute value of couplings or biases.

4.6.5 QAPLIB

QAPLIB [50] provides a collection of QAPs together with the best known solution. This is the most commonly used benchmark for QAPs, with sets of problems from a variety of settings, *e.g.* graphs but also unstructured data. For some instances it is unclear if the best known solution is actually the optimum due to the complexity of the problem.

We ran cyclic α-expansion on some of the subsets ensuring that every 2-cycle occurs once when optimizing, and repeat this three times. We always choose the best solutions from 5000 anneals if the size is > 25, and 500 anneals otherwise. The relative error we achieved is in Fig. 4.6. For instances from [51, 99, 119], our solution is almost always within 1% of the best solution. For the $n = 16$ instances in [99], we achieved the optimal solution in 9 out of 10 cases. Other sets contain instances up to size 30 and pose a bigger challenge, our worst result was within 15% of the optimum. We report our exact solutions in the supplementary material.

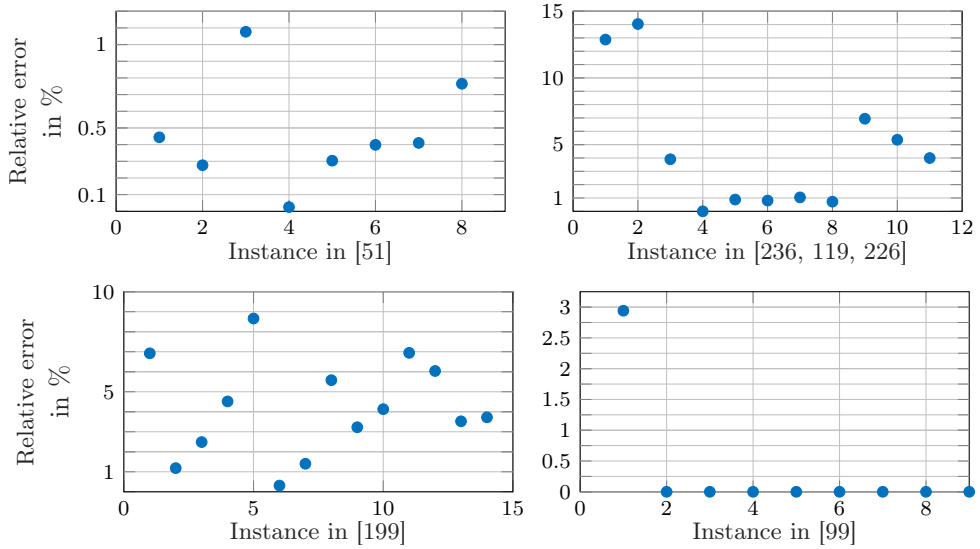


Figure 4.6: Relative error $\frac{E_{\text{obtained}}-E_{\text{opt}}}{E_{\text{opt}}}$ of our method in percentage for the instances of [51], [236] (1-3), [119] (4-8) and [226] (9-11),[199], and [99] in QAPLIB. The problem sizes range between 12 and 30, of which [199] contains the larger ones where we do less well.

4.7 Conclusion

We introduce Q-Match and experimentally show that it can solve correspondence problems of sizes encountered in practical applications. This conclusion is made for the first time in literature for methods utilizing quantum annealing. Refraining from explicit permutation constraints and addressing the problem iteratively allowed us to significantly outperform the previous quantum state of the art, both in terms of the supported problem sizes and the probability to measure a globally optimal solution. We even achieved results comparable to a classical method. We hope that our work inspires further research in quantum annealing and related optimization problems in the vision community.

Acknowledgements. This work was partially supported by the ERC consolidator grant 4DReply (770784).

4.8 Supplementary Material for Q-Match: Iterative Shape Matching via Quantum Annealing

This supplementary material provides a deeper analysis of the proposed Q-Match approach and more experimental details. This includes:

- Further analysis of the solution quality for individual QUBOs and visualizations of the minor embeddings (Sec. 4.9).
- Derivation of Eq. (4.6) and proof of Lemma 4.1 from the main matter (Sec. 4.10).
- A toy example, in which all 2-cycles individually do not improve the energy (Sec. 4.11).

- More details on calculating W_s and $\tilde{W}$ (Sec. 4.12).
- A list of our solutions to selected QAPLIB problems *vs* ground truth (Sec. 4.13).

4.9 Analysis of the Individual QUBOs

We analyze the quality of the solution of the individual QUBOs in dependence of the dimension. The success probability for one QUBO solution is defined as the fraction between the anneals that ended up in the optimum and the number of anneals. The success probability averaged over 20 QUBOs per dimension at the first two iterations, *i.e.,* computations of (4.5) for a set of 2-cycles, of one instance of the FAUST dataset can be seen in Fig. 4.7. We see that for increasing dimension the success probability is decreasing and less runs end up in the ground state. One possible way to reverse this trend would be to increase the annealing time, which we left constant at $T_A = 20\mu s$. We also plotted the fraction of QUBOs, where the ground state was among the returned solutions. Leaving the number of anneals constant this probability declined from 1 for $4 - 24$ worst vertices to 0.4 for 40 worst vertices. To get the optimum for more instances we performed the experiment with 5000 anneals for 40 and 50 worst vertices. In this experiment we found the optimum in 90% or 45% of the cases, respectively.

Note that quantum annealing is a stochastic algorithm. Therefore a success probability P_s is directly linked with the amount of time needed to to get the optimum with, *e.g.,* 99% probability:

$$T = \frac{\ln(1 - 99\%)}{\ln(1 - P_s)} T_A, \tag{4.11}$$

where T_A is the time for one anneal. We also computed the binomial proportion confidence interval for the probability that the optimum of the QUBO is found, with at least one anneal. If the underlying distribution is binomial, then the true probability lies within 20% of our estimates in 95% of cases. If the number of worst vertices is 23 or less the estimate is even closer to the true probability. The binomial confidence interval can be seen in Fig. 4.7, on the right. In the left plot of Fig. 4.7 the standard deviation of the success probability for averaging over the 20 individual QUBOs is also depicted. This shows that the success probability strongly varies for the distinct QUBO instances.

4.9.1 Minor Embedding

In most cases, our problems result in fully connected logical qubit graphs. In Figs. 4.8 and 4.9, there are illustrations of the minor embeddings computed by the method of Cai *et al.* [52] (used in Leap 2) and visualized by the problem inspector of Leap 2 [74]. We plot the average maximum chain lengths and average numbers of physical qubits in the obtained minor embeddings in Fig. 4.10.

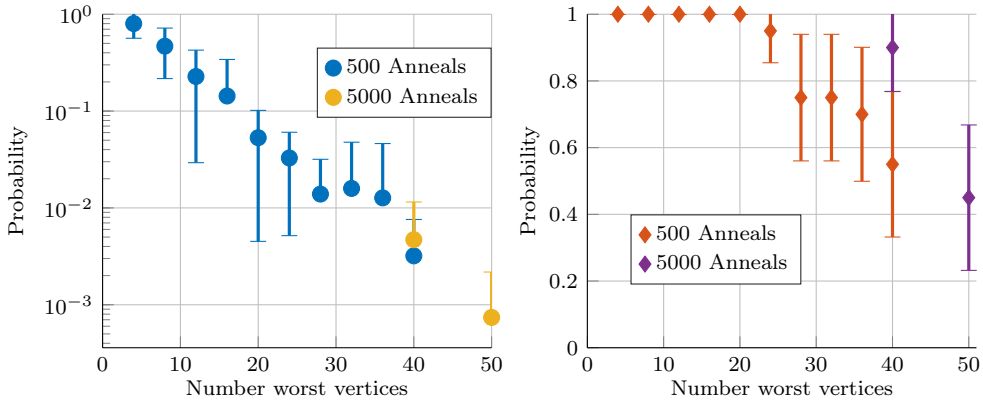


Figure 4.7: Success probability (left) and the fraction of executions where the best solution is the optimum (right), for different problem sizes and number of anneals. The success probability decreases with the problem size, and, therefore, more anneals are necessary. In the left plot the deviation bar is the standard deviation and in the right plot it is the binomial proportion confidence interval. No lower part of the deviation bar means it goes beyond zero in the left plot.

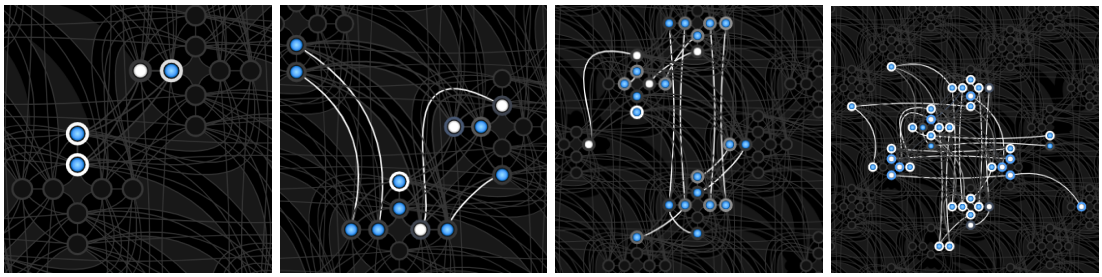


Figure 4.8: Illustration of the embedding from the D-Wave Leap 2 problem inspector [74] for using $8, 16, 24, 32$ worst vertices. One node depicts a physical qubits. The inner color shows the measured value in the lowest energy state, while the color of the outer ring shows the sign of the bias, *e.g.,* the coefficient of the linear term in the optimization problem (4.2). The edges depict the chains.

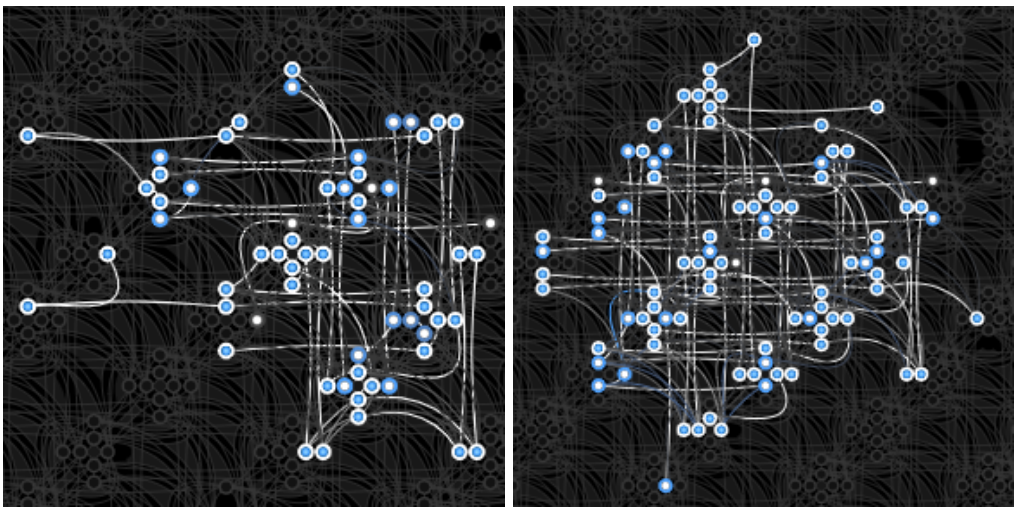


Figure 4.9: Illustration of the embedding from the D-Wave Leap 2 problem inspector [74] for using 40 and 50 worst vertices. One node depicts a physical qubits. The inner color shows the measured value in the lowest energy state, while the color of the outer ring shows the sign of the bias, *e.g.*, the coefficient of the linear term in the optimization problem (4.2). The edges depict the chains.

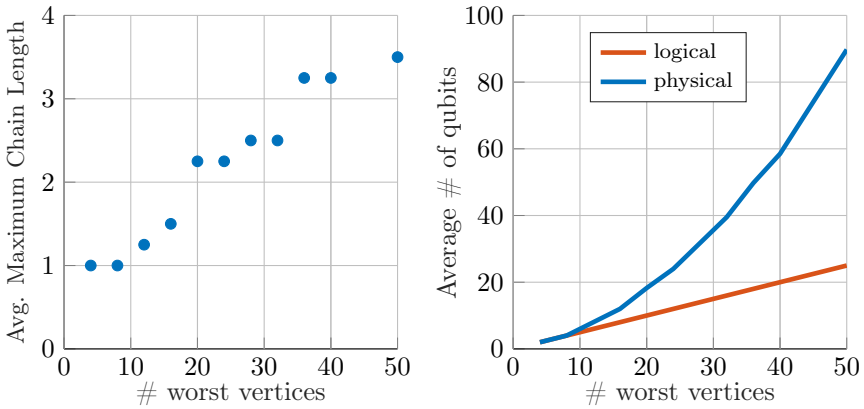


Figure 4.10: Average maximal chain length and number of physical qubits for increasing problem size averaged over 4 instances. The number of logical qubits increases linearly with the problem size.

4.10 Derivations and Proofs

4.10.1 Derivation of Eq. (4.6) in Sec. 4.5.1

It is stated in the main paper in Eq. (4.6), that one can convert the multiplication of cycles from (4.5) into an additive structure, *i.e.,* that

$$P(\alpha) = \left(\prod_i c_i^{\alpha_i}\right) P_0 = P_0 + \sum_{i=1}^{m} \alpha_i (c_i - I) P_0$$

holds, where I is the identity.

Proof. Consider the case where we only have a single cycle c. Now, the following holds:

$$\begin{aligned} P(\alpha) &= P_0 + \alpha(c - I)P_0 = (1-\alpha)P_0 + \alpha c P_0 \\ &= \begin{cases} P_0, & \text{for } \alpha = 0 \\ cP_0, & \text{for } \alpha = 1 \end{cases} \;=\; c^\alpha P_0. \end{aligned}$$

Independent of P_0, we can write:

$$c^\alpha = (1-\alpha)I + \alpha c. \tag{4.12}$$

Additionally we can write (4.6) independent of P_0 by applying the inverse permutation from the right side:

$$\prod_i c_i^{\alpha_i} = I + \sum_{i=1}^{m} \alpha_i (c_i - I).$$

Now as an induction step we apply $c_{m+1}^{\alpha_{m+1}}$ from the right:

$$\begin{aligned} & c_{m+1}^{\alpha_{m+1}} \left(\prod_i c_i^{\alpha_i}\right) \\ &= c_{m+1}^{\alpha_{m+1}} (I + \sum_{i=1}^{m} \alpha_i (c_i - I)) \\ &= ((1-\alpha_{m+1})I + \alpha_{m+1} c_{m+1})(I + \sum_{i=1}^{m} \alpha_i (c_i - I)) \\ &= I + \sum_{i=1}^{m+1} \alpha_i (c_i - I) + \alpha_{m+1}(c_{m+1} - I) \sum_{i=1}^{m} \alpha_i (c_i - I). \end{aligned} \tag{4.13}$$

We want to use that for two disjoint cycles c_k, c_l, the equality $(c_k - I)(c_l - I) = 0$ holds. In all the places where c_k has 0 on the diagonal, c_l has 1, because they are disjoint. This leads to the fact that in the rows, where $(c_k - I)$ is non-zero, $(c_l - I)$ has zero columns or rows, respectively. Therefore the last term in (4.13) vanishes and the statement is proven.

□

4.10.2 Proof of Lemma 4.1

To prove Lemma 4.1, we first show that the statement is correct for a k-cycle.

Lemma 4.2. *Let P be a k-cycle. Then, P can be written as a product of Q and R, where Q and R are permutations that only consist of disjoint 2-cycles.*

Proof. Without loss of generality, let $P = (1\ 2\ 3...k)$ be the k-cycle. This is possible because rearranging rows and columns does not change the problem. For even k, Q and R take the following form:

$$Q = (1\ 2)(3\ k)(4\ (k-1))\dots\left(\left(1+\frac{k}{2}\right)\ \left(2+\frac{k}{2}\right)\right),$$
$$R = (2\ k)(3\ (k-1))\dots\left(\frac{k}{2}\ \left(2+\frac{k}{2}\right)\right).$$

It can be easily checked that $P = QR$ holds in this case. For uneven k, one can choose the following Q and R, and the same holds:

$$Q = (1\ 2)(3\ k)(4\ (k-1))\dots\left(\left(\frac{k+1}{2}\right)\ \left(1+\frac{k+1}{2}\right)\right),$$
$$R = (2\ k)(3\ (k-1))\dots\left(\frac{k-1}{2}\ \left(1+\frac{k+1}{2}\right)\right).$$

□

Next, the following argument gives the proof for Lemma 4.1.

Proof. One can first write the permutation P in cycle notation. Then, we decompose each cycle individually as it was shown in Lemma 4.2. Note that according to Lemma 4.2, the decomposition of the k-cycle does not require additional elements in Q or R that do not occur in the cycle. □

Permutations like Q or R that only consist of 2-cycles are called involutions. The fraction of permutations that are involutions has following asymptotic behavior [103] (Prop. VIII.2.):

$$\frac{I_n}{n!} = \frac{e^{-\frac{1}{4}}}{2\sqrt{\pi n}} n^{-\frac{n}{2}} e^{\frac{n}{2}+\sqrt{n}} \left(1 + O\left(\frac{1}{n^{\frac{1}{5}}}\right)\right). \tag{4.14}$$

Considering Lemma 4.2, a rough estimate is $I_n^2 > n!$. For these permutations, in theory, one step of the cyclic α-expansion would suffice to obtain to the identity (which could be the optimum w.l.o.g.).

Note on Classical Optimization: If (4.8) was submodular, it would be possible to efficiently solve it by converting it to a graph cut problem instead of using AQCing. According to Bach [24], a quadratic function $f(\mathbf{x}) := \mathbf{x}^T W \mathbf{x}$ is submodular, if and only if all non-diagonal elements of W are non-positive. Since $\tilde{W}_{ij}$ can have a positive sign, the function is not submodular and, therefore, cannot be efficiently optimized. We tried small alternations of $\tilde{W}$ (*e.g.*, changing the sign of the off-diagonal elements by switching C_i to $-C_i$), but did not succeed in making (4.8) submodular.

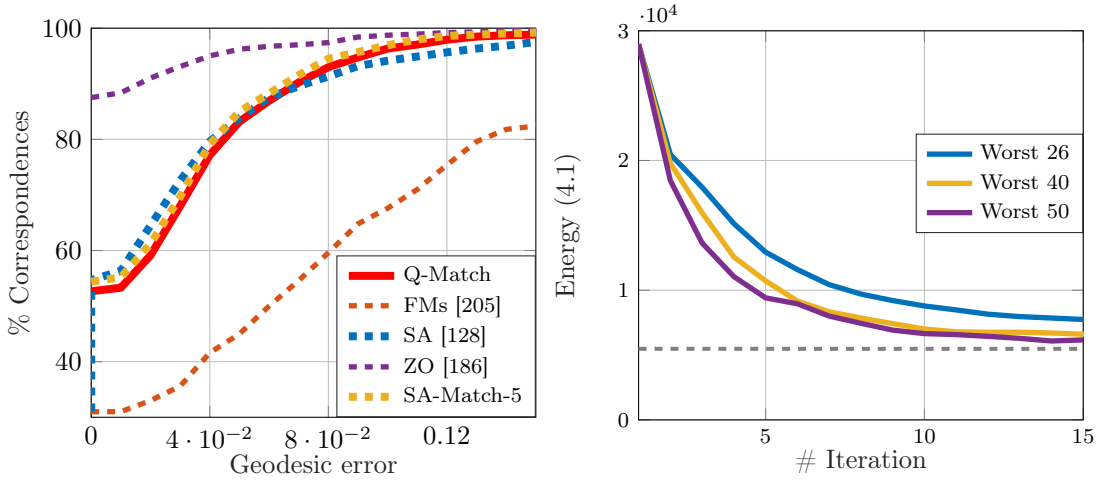


Figure 4.11: Evaluation of cumulative error [144] (left) and convergence (right) on the FAUST dataset. (Left) We compare against Simulated Annealing [128] without postprocessing and Functional Maps [205]. Dashed lines indicate non-quantum approaches. The results have symmetry-flipped solutions removed, these have an equivalent final energy for all three methods but are not recognized as correct in the evaluation protocol. (Right) We show the convergence of the energy over 30 iterations. The larger the set of worst vertices, the faster our method converges. The dashed grey line shows the optimal energy.

4.10.3 Simulated Annealing vs. Quantum Annealing for the Subproblems

We also performed Q-Match from Alg. 2 with a simulated annealing solver from [75] for the subproblems. The quality of the results for the FAUST dataset and for QAPLIB can be seen in Figs. 4.11 and 4.12. Here we executed the simulated annealing sampler with 5 sweeps and performed 100 runs. Increasing the number of sweeps further did not yield qualitatively different results.

In Fig. 4.13 the processing time for the subproblems is plotted in dependence of the dimension. If one measures the wall clock time for a query to the D-Wave sampler in the same way one gets results of the order of seconds. This is due to the fact that the time the solver takes for the computation is overshadowed by the time to connect and get access to the machine. To get this time estimate one can simply look at `dwave ping` in ocean. For the QPU access time we already presented the results in dependence of dimension in Fig. 4.5. A more detailed overview of the different runtimes is given in [76].

Since the minor embedding is computed locally one can directly measure the processing time in dependence of the dimension. In Fig. 4.14 this measurement is averaged over 5 instances. The simulated annealing sampler takes here already an order of magnitude less time. However since we mostly want to embed a fully connected graph the problem to find an embedding could be solved beforehand and an existing embedding can be reused.

We also state that the emphasis of this work is not on benchmarking the subproblems. If one would do this one could also optimize over the annealing path and could use further features like the extended J-range for more precision and spin reversal transformations to mitigate some

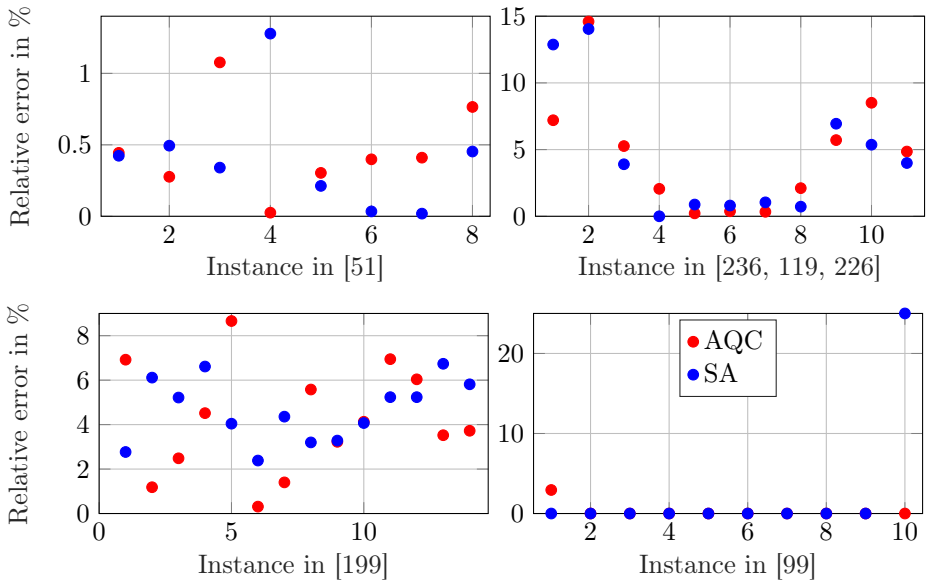


Figure 4.12: Relative error $\frac{E_{\text{obtained}}-E_{\text{opt}}}{E_{\text{opt}}}$ of our method in percentage for the instances of [51] (upper left), [236] (1-3), [119] (4-8) and [226] (9-11) (upper right), [199] (lower left) and [99] (lower right) in QAPLIB. The problem sizes range between 12 and 30, of which [199] contains the larger ones where we do less well.

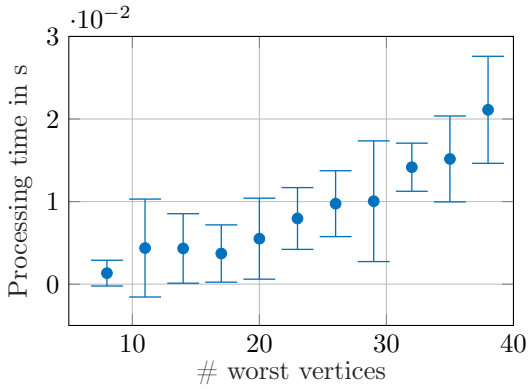


Figure 4.13: Processing time of the simulated annealing sampler in dependence of the dimension. For one execution we have 100 runs and 5 sweeps as parameters. We averaged the measured time over 10 executions.

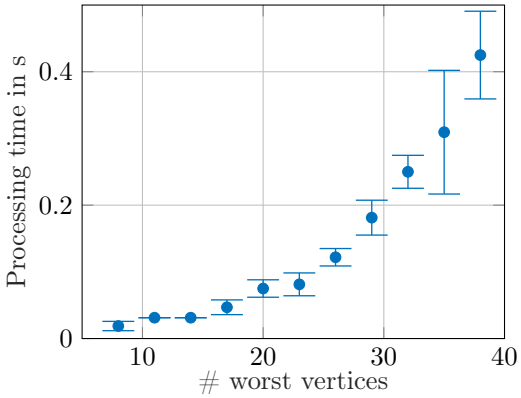


Figure 4.14: Time to compute the minor embedding in dependence of the dimension of the subproplem. The subproblems stem from the Q-Match algorithm applied to the Faust dataset. We averaged the measured time over 5 executions.

systematic errors. Additionally one would also apply formula (4.11). In this section we only want to provide a rough overview of the computing time. Preliminary work that does an in depth optimization of these algorithms confirms that quantum annealing is highly promising: In [86] QUBOs were found where quantum annealing yields a "time-to-99%- success-probability that is $\sim 10^8$ times faster than simulated annealing running on a single processor core". Reference [138] benchmarks the D-Wave 2000Q on a broader class of problems and confirms the potential of this technology.

4.11 Failure Case for Individually Optimizing Over the 2-Cycles

We present an example to proof that optimizing over larger sets of 2-cycles is superior to looking at single 2-cycles separately, as is done in [128]. We construct a plane 2D-shape where the optimum can be reached with a collection of 2-cycles but each 2-cycle applied individually results in a worse energy starting from a specific permutation. Consider the points depicted in Figure 4.15. If we chose a, b and ϵ, we can compute the energies of permutations with (4.9) using Euclidean distances between the points. Possible values would be $b = 10$, $a = 1$ and $\epsilon = 0.1$.

The shape is almost (but not exactly) symmetric with respect to mirroring along a shifted x-axis (Permutation (1 4)(2 5)(3 6)). In our experiment, the second shape is a copy of the first with permuted vertices, and we want to find the correspondence. Let the identity be the optimal solution, and the current permutation is $P_0 = (14)(25)(36)$. Permuting any of the three points on the upper $\{1, 2, 3\}$ to any of the lower points $\{4, 5, 6\}$ on the right causes – despite the correct assignment – a distortion of the (near-) isometry, such that no such 2-cycle improves the cost function when the assignment of the other points remains unchanged. However, applying all three correct 2-cycles at once, allows to pass to the global optimum with a lower energy.

This illustrates that using our cyclic α-expansion iteration step for optimizing over multiple 2-cycles at a time can have significant advantages over a sequence of simple single 2-cycle updates.

BO [51]:

	a	b	c	d	e	f	g	h
Optimum	5426670	3817852	5426795	3821225	5386879	3782044	10117172	7098658
Our (Quantum)	5450757	3828405	5485230	3822190	5403238	3797120	10158673	7152966
Our (Simulated Annealing)	5449653	3836716	5445272	3870067	5398333	3783329	10119061	7130826

ESC16 [99], HAD [119]:

	a	b	c	d	e	f	g	h	i	j	a	b	c	d	e
Optimum	68	292	160	16	28	0	26	996	14	8	1652	2724	3720	5358	6922
Our (Quantum)	70	292	160	16	28	0	26	996	14	8	1652	2748	3750	5358	6922
Our (Simulated Annealing)	68	292	160	16	28	0	26	996	14	10	1686	2730	3734	5376	7068

NUG [199]:

	a	b	c	d	e	f	g	h	i	j	k	l	m	n
Optimum	578	1014	1610	1240	1732	1930	2570	2438	3596	3488	3744	5234	5166	6124
Our (Quantum)	618	1026	1650	1296	1882	1936	2606	2574	3712	3632	4004	5550	5348	6352
Our (Simulated Annealing)	594	1076	1694	1322	1802	1976	2682	2516	3714	3630	3940	5508	5514	6480

SCR [236], Rou [226]:

	a	b	c	a	b	c
Optimum	31410	51140	110030	235528	354210	725522
Our (Quantum)	35454	58320	114322	251872	373218	754506
Our (Simulated Annealing)	33672	58606	115822	248982	384354	760738

Table 4.2: Our solutions for exemplary sets of the QAPLIB dataset with different sizes of quadratic assignment problems.

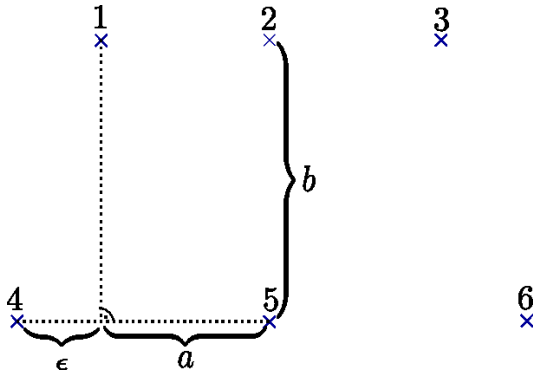


Figure 4.15: Exemplary shape to show that individually applying 2-cycles does not suffice. Because of the length ϵ the points are only nearly symmetric along an x-axis. The shape is invariant under the permutation (13)(46).

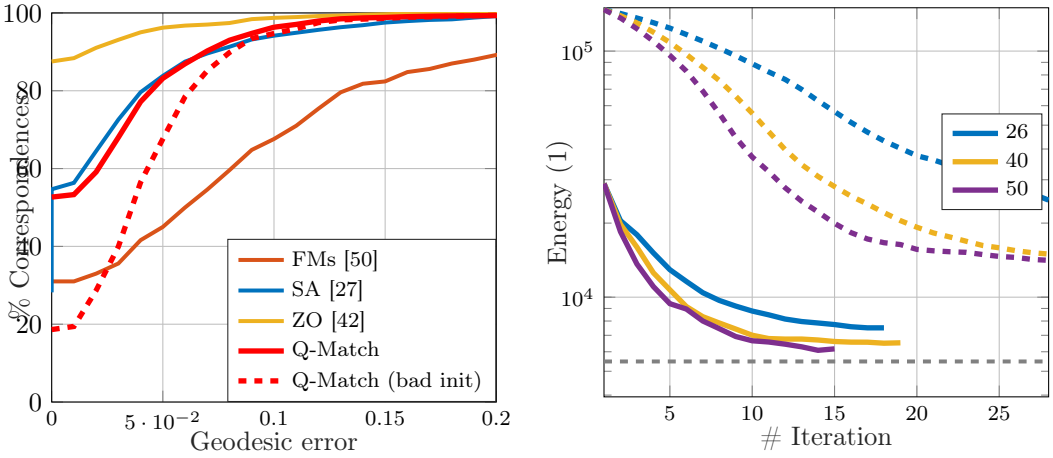


Figure 4.16: Quantitative experiments comparing our method using the optimal descriptor initialization (solid lines) with the worst descriptor initialization (dashed lines). (Left) Cumulative geodesic error (left) and convergence (right) is shown on the FAUST dataset, otherwise equivalent to Fig. 4.3. The dashed gray line is the ground truth value.

4.12 Calculation of W_s and $\tilde{W}$

Notice that for a given permutation P and a set of cycles C, it is possible to get $\tilde{W}$ without precomputing W_s in roughly the same time as computing W_s. However, if W_s is precomputed for a subset of vertices, $\tilde{W}$ can be computed very efficiently for any set of cycles on this subset. Therefore, we once calculate the expensive W_s, and then evaluate several sets of cycles on it to increase the overall efficiency.

4.12.1 Calculating W_s

If we want to solve a subproblem of (4.1) we assume that all correspondences for indices, that are not optimized, stay fixed. Therefore, it is not sufficient to set W_s to a submatrix of W, but we have to add the influence of these fixed correspondences. Given a set $s_M, s_N \subset \{1, \dots, n\}$ of indices which indicate the subproblem of W (in Q-Match, these are the sets I_M, I_N), and a previous permutation P, we calculate W_s as follows:

$$(W_s)_{ikjl} = W_{ikjl} + \sum_{(v_M, v_N) \in V} W_{ikv_M v_N} + W_{v_M v_N jl}. \tag{4.15}$$

Here $V \subset M \times N$ is the set of correspondences indicated by a permutation P, with removed all tuples in V which contain entries from s_M, s_N, *i.e.*, $(v_M, v_N) \in V$ if $P(v_M) = v_N$ and $v_M \notin v_M, v_N \notin v_N$. This results in a $k^2 \times k^2$ matrix where each entry contains the sum of $\mathcal{O}(|C|)$ basic operations ($W_{ijkl} = |d(i,j) - d(k,l)|$, where all $d(\cdot,\cdot)$ are precomputed), resulting in $\mathcal{O}(k^4|C|)$. The computation of each entry can be parallelized.

4.12.2 Calculating $\tilde{W}$

Since we converted (4.1) into a QUBO (4.2), W_s also needs to be converted into $\tilde{W}$, *i.e.,* the matrix describing the energy for the chosen combination of cycles. Since the cycles are sampled from s_M, s_N, $\tilde{W}$ can be computed from the entries of W_s, as we defined in (7) (and repeated here):

$$\tilde{W}_{ij} = \begin{cases} E(C_i, C_j) & \text{if } i \neq j, \\ E(C_i, C_i) + E(C_i, P_0) + E(P_0, C_j) & \text{otherwise.} \end{cases} \tag{7}$$

$E(C_i, C_j)$ can be calculated as two matrix-vector multiplications (4.1), however, since the vectors are vectorized permutation matrices with exactly k non-zero entries, they can be written as two sums over k entries. This is a $m \times m$ matrix. Computing every entry separately leads to a complexity of $\mathcal{O}(m^2 k)$. In our setting with 2-cycles, $m = \frac{1}{2}k$ holds, therefore, we reach a complexity of $\mathcal{O}(k^3)$. Note that usually $|C| \gg k$, and calculating $\tilde{W}$ is a lot more efficient than calculating W_s (see Fig. 4.5).

4.13 Exact Solutions on QAPLIB

Since the relative error of the QAP is not invariant under shifts of W, we also report our results on QAPLIB in Table 4.2. Here, it becomes clear again that we reach the optimum for virtually all instances in ESC16 and HAD.

4.14 Non-Optimal Initialization

Due to a sign error in our original experiments on the FAUST dataset, we ran them with the worst possible descriptor based initialization instead of the best. As expected the accuracy is not as high and the algorithm converges slower, but Q-Match does not break completely with a very bad initialization. We see this as an indicator that finding high quality solutions for larger subproblems leads to a very robust pipeline. See Figure 4.16 for the exact results.

Chapter 5

CCuantuMM: Cycle-Consistent Quantum-Hybrid Matching of Multiple Shapes

5.1 Contextualization

This chapter is a reprint of the CVPR publication "CCuantuMM: Cycle-Consistent Quantum-Hybrid Matching of Multiple Shapes". In this work, the publication from the previous chapter, Q-Match, is generalized to the multi-shape setting. Empirically, we observed that considering three shapes at a time gives already good results. The methodology we used to iteratively optimize the matching for N shapes could even be of general interest in shape matching without considering only quantum annealing application scenarios.

Zorah Lähner and Michael Möller had the idea of investigating the application of quantum annealing for the multi-shape matching problem. Harshil Bhatia performed experiments for the final publication. Marcel Seelbach Benkner performed experiments at an earlier stage and later helped in the discussions and writing. Edith Tretschk significantly contributed to the design of the experiments, the writing and via active discussion. Zorah Lähner again provided shape matching expertise, which made the work possible. Michael Möller and Vladislav Golyanik contributed a lot with supervision and played a large role in planning the experiments. Christian Theobalt provided helpful feedback.

Abstract

Jointly matching multiple, non-rigidly deformed 3D shapes is a challenging, $\mathcal{NP}$-hard problem. A perfect matching is necessarily cycle-consistent: Following the pairwise point correspondences along several shapes must end up at the starting vertex of the original shape. Unfortunately, existing quantum shape-matching methods do not support multiple shapes and even less cycle consistency. This paper addresses the open challenges and introduces the first quantum-hybrid approach for 3D shape multi-matching; in addition, it is also cycle-consistent. Its iterative



Figure 5.1: **Our quantum-hybrid method matches all 100 shapes of the FAUST collection [41] with guaranteed cycle consistency (white arrows).** Here, we visualise the matchings via texture transfer between all shapes. Our method scales linearly in the number of shapes. See the full figure in the supplement.

formulation is admissible to modern adiabatic quantum hardware and scales linearly with the total number of input shapes. Both these characteristics are achieved by reducing the N-shape case to a sequence of three-shape matchings, the derivation of which is our main technical contribution. Thanks to quantum annealing, high-quality solutions with low energy are retrieved for the intermediate $\mathcal{NP}$-hard objectives. On benchmark datasets, the proposed approach significantly outperforms extensions to multi-shape matching of a previous quantum-hybrid two-shape matching method and is on-par with classical multi-matching methods. Our source code is available at `4dqv.mpi-inf.mpg.de/CCuantuMM/`.

5.2 Introduction

Recently, there has been a growing interest in applying quantum computers in computer vision [40, 111, 185]. Such quantum computer vision methods rely on quantum annealing (QA) that allows to solve $\mathcal{NP}$-hard quadratic unconstrained binary optimisation problems (QUBOs). While having to formulate a problem as a QUBO is rather inflexible, QA is, in the future, widely expected to solve QUBOs at speeds not achievable with classical hardware. Thus, casting a problem as a QUBO promises to outperform more unrestricted formulations in terms of tractable problem sizes and attainable accuracy through sheer speed.

A recent example for such a problem is shape matching, where the goal is to estimate

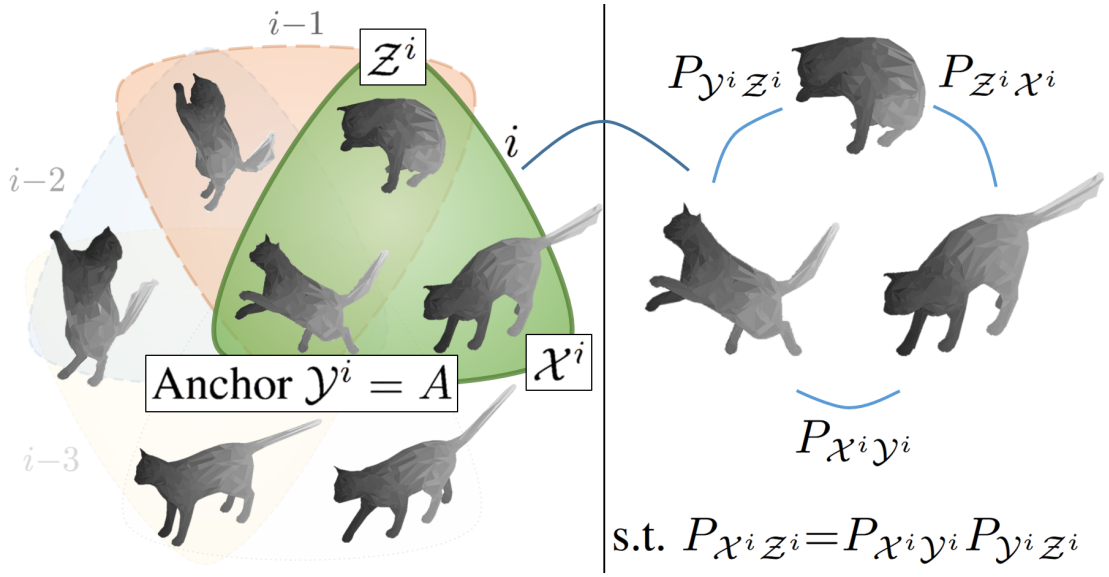


Figure 5.2: We match N shapes by iteratively matching triplets.

correspondences between two shapes. Accurate shape matching is a core element of many computer vision and graphics applications (*i.e.,* texture transfer and statistical shape modelling). If non-rigid deformations are allowed, even pairwise matching is $\mathcal{NP}$-hard, leading to a wide area of research that approximates this problem, as a recent survey shows [87]. Matching two shapes is one of the problems that was shown to benefit from quantum hardware: Q-Match [238] iteratively updates a subset of point correspondences using QA. Specifically, its cyclic α-expansion allows to parametrise changes to permutation matrices without relaxations.

The question we ask in this work is: How can we design a *multi-shape* matching algorithm in the style of Q-Match that has the same benefits? As we show in the experiments, where we introduce several naïve multi-shape extensions of Q-Match, this is a highly non-trivial task. Despite tweaking them, our proposed method significantly outperforms them.

If $N{>}2$ shapes have to be matched, the computational complexity of naïve exhaustive pairwise matching increases quadratically with N, which does not scale to large N. Furthermore, these pairwise matchings can easily turn out to be inconsistent with each other, thereby violating cycle consistency. For example, chaining the matchings $P_{\mathcal{XY}}$ from shape $\mathcal{X}$ to $\mathcal{Y}$ and $P_{\mathcal{YZ}}$ from $\mathcal{Y}$ to $\mathcal{Z}$ can give very different correspondences between $\mathcal{X}$ and $\mathcal{Z}$ than the direct, pairwise matching $P_{\mathcal{XZ}}$ of $\mathcal{X}$ and $\mathcal{Z}$: $P_{\mathcal{XZ}} \neq P_{\mathcal{XY}}P_{\mathcal{YZ}}$. (We apply the permutation matrix $P_{\mathcal{XY}}$ to the one-hot vertex index vector $x \in \mathcal{X}$ as $x^\top P_{\mathcal{XY}} = y \in \mathcal{Y}$.) Thus, how can we achieve cycle consistency by design? A simple solution would be to match a few pairs in the collection to create a spanning tree covering all shapes and infer the remaining correspondences by chaining along the tree. Despite a high accuracy of methods for matching two shapes, this correspondence aggregation policy is prone to error accumulation [229]. A special case of this policy is pairwise matching against a single anchor shape, which also guarantees cycle-consistent solutions by construction [109]. We build on this last option in our method as it avoids error accumulation.

This paper, in contrast to purely classical methods, leverages the advantages of quantum

computing for multi-shape matching and introduces a new method for simultaneous alignment of multiple meshes with guaranteed cycle consistency; see Fig. 5.1. It makes a significant step forward compared to Q-Match and other methods utilising adiabatic quantum computing (AQC), the basis for QA. Our cycle-consistent quantum-hybrid multi-shape matching (CCuantuMM; pronounced "quantum") approach relies on the computational power of modern quantum hardware. Thus, our main challenge lies in casting our problem in QUBO form, which is necessary for compatibility with AQC. To that end, two design choices are crucial: (1) Our method reduces the N-shapes case to a series of three-shape matchings; see Fig. 5.2. Thus, CCuantuMM is iterative and hybrid, *i.e.,* it alternates in every iteration between preparing a QUBO problem on the CPU and sampling a QUBO solution on the AQC. (2) It discards negligible higher-order terms, which makes mapping the three-shape objective to quantum hardware possible. In summary, the core technical contributions of this paper are as follows:

- CCuantuMM, *i.e.,* a new quantum-hybrid method for shape multi-matching relying on cyclic α-expansion. CCuantuMM produces cycle-consistent matchings and scales linearly with the number of shapes N.
- A new formulation of the optimisation objective for the three-shapes case that is mappable to modern QA.
- A new policy in shape multi-matching to address the N-shape case relying on a three-shapes formulation and adaptive choice of an anchor shape.

Our experiments show that CCuantuMM significantly outperforms several variants of the previous quantum-hybrid method Q-Match [238]. It is even competitive with several non-learning-based classical state-of-the-art shape methods [186, 109] and can match more shapes than them. In a broader sense, this paper demonstrates the very high potential of applying (currently available and future) quantum hardware in computer vision.

5.3 Related Work

Quantum Computer Vision (QCV). Several algorithms for computer vision relying on quantum hardware were proposed over the last three years for such problems as shape matching [111, 238, 185], object tracking [163, 275], fundamental matrix estimation, point triangulation [90] and motion segmentation [18], among others. The majority of them address various types of alignment problems, *i.e.,* transformation estimation [111, 185], point set [111, 198] and mesh alignment [238], graph matching [237, 40] and permutation synchronisation [40].

Only one of them, QSync [40], can operate on more than two inputs and ensure cycle consistency for the underlying matchings. In contrast to QSync, we can align inputs with substantially larger (by two orders of magnitude) shapes in the number of vertices. Furthermore, we address a different problem, *i.e.,* mesh alignment, for which an algorithm for two-mesh alignment with the help of AQC exists, namely Q-Match [238], as we discuss in the introduction.

To maintain the valid structure of permutation matrices, Quantum Graph Matching, QGM[237] and Q-Sync [40] impose linear constraints. However, this requires that the corresponding penalty parameter is carefully chosen. If the parameter is chosen too big and the linear constraints are enforced too strongly, this severely limits QGM and Q-Sync's ability to handle large sets of vertices. On the other hand, if the linear constraints are enforced too weakly, there is no guarantee

to obtain valid permutations as solutions. As discussed in the introduction, our approach follows Q-Match to ensure valid permutation matrices by construction.

Multi-Shape Matching. We focus this section on multi-shape and non-learning methods as CCuantuMM falls in this category. As our approach is not learning-based, it trivially generalises to unknown object categories without a need for training data. For a general survey of recent advances in shape matching, see Sahillioglu [230].

Matching shape pairs is a classical problem in geometry processing [186]. When more than two shapes of the same class exist, stronger geometric cues can be leveraged to improve results by matching all of them simultaneously. Unfortunately, the already very high problem complexity increases even further the more shapes are used. Hence, existing multi-shape matching methods limit the total number of shapes and their resolution [66, 109], work in spectral space [131], or relax the permutation constraints [140]. Early multi-matching methods computed pair-wise matchings and subsequently used permutation synchronisation to establish cycle consistency [207, 178, 239]. Still, permutation synchronisation requires the eigendecomposition of a matrix with quadratically increasing dimensions [207].

HiPPI [34] is a computationally efficient method that takes geometric relations into account while generalising permutation synchronization but is still limited in resolution. Instead of looking at permutations directly, ZoomOut [186] reduces the dimensionality of the problem by projecting it onto the spectral decomposition. This idea has been extended to take cycle consistency within the spectral space into account [130], which does not guarantee a point-wise consistent matching. To circumvent this issue, IsoMuSh [109] jointly optimises point and functional correspondences. The method detangles the optimisation into smaller subproblems by using a so-called *universe shape* that all shapes are mapped to instead of each other, as Cao and Bernard do [54]. Using a *universe* is similar to requiring a template shape, as many learning-based approaches do [115, 89, 250]: Both synchronise all correspondences by matching them through a unified space. This is similar to the concept of anchor shape we use but inherently less flexible because the universe size or template have to be given *a priori*. Our anchor is chosen from the given collection as part of the method. Although using an anchor slightly improves our results, we note that our method does not necessarily require one for operation. Hence, a random shape could be picked instead in each iteration without an increase in complexity if using an anchor is not feasible or does not represent the shape collection well.

5.4 Background

5.4.1 Adiabatic Quantum Computing (AQC)

AQC is a model of computation that leverages quantum effects to obtain high-quality solutions to the $\mathcal{NP}$-hard problem class of Quadratic Unconstrained Binary Optimisation (QUBO) problems: $\min_{x\in\{0,1\}^k} x^T Q x$, for $k \in \mathbb{N}$ and a QUBO weight matrix $Q \in \mathbb{R}^{k\times k}$. Each entry of x corresponds to its own logical qubit, the quantum equivalent of a classical bit. The diagonal of Q consists of linear terms, while the off-diagonals are inter-qubit *coupling weights*. A QUBO can be classically tackled with simulated annealing (SA) [262] or a variety of other discrete optimisation techniques [151, 95], which, for large k, typically yield only approximate solutions as QUBOs are in general $\mathcal{NP}$-hard. AQC holds the potential to systematically outperform classical approaches such as SA, see [86, 147] for an example. AQC exploits the adiabatic theorem of quantum mechanics

[44]: If, when starting from an equal superposition state of the qubits (where all solutions $\{0,1\}^k$ have the same probability of being measured) and imposing external influences corresponding to the QUBO matrix on the qubits sufficiently slowly (called *annealing*), they will end up in a quantum state that, when measured, yields a minimizer x of the QUBO. Not all physical qubits on a real quantum processing unit (QPU) can be connected (coupled) with each other. Thus, a *minor embedding* of the logical-qubit graph (defined by non-zero entries of the QUBO matrix) into the physical-qubit graph (defined by the hardware) is required [52]. This can lead to a chain of multiple physical qubits representing a single logical qubit. For details of quantum annealing on D-Wave machines, we recommend [182].

5.4.2 Shape Matching

The problem of finding a matching for non-rigidly deformed shapes having n vertices can be formulated as an $\mathcal{NP}$-hard Quadratic Assignment Problem (QAP) [140, 49]:

$$\min_{P \in \mathbb{P}_n} p^T W p, \tag{5.1}$$

where $p = \mathrm{vec}(P) \in \{0,1\}^{n^2}$ is a flattened permutation matrix, and $W \in \mathbb{R}^{n^2 \times n^2}$ is an energy matrix describing how well certain pairwise properties are conserved between two pairs of matches. If two shapes $\mathcal{X}, \mathcal{Y}$ are discretised with n vertices each, W is often chosen as [140]:

$$W_{x_1 \cdot n + y_1, x_2 \cdot n + y_2} = \| d^g_{\mathcal{X}}(x_1, x_2) - d^g_{\mathcal{Y}}(y_1, y_2) \|, \tag{5.2}$$

where x_1, x_2 are vertices on $\mathcal{X}$; y_1, y_2 are vertices on $\mathcal{Y}$; and $d^g_{\mathcal{I}}(\cdot,\cdot)$ is the geodesic distance on the shape $\mathcal{I}$. Therefore, $W_{x_1 \cdot n + y_1, x_2 \cdot n + y_2}$ represents how well the geodesic distance is preserved between corresponding pairs of vertices on the two shapes. Instead of pure geodesics, Gaussian-filtered geodesics are also a popular choice for W [265]:

$$g_{\mathcal{X}}(x_1, x_2) = \frac{1}{\rho\sqrt{2\pi}} \exp\left\{ \left(-\frac{1}{2} \left(\frac{d^g_{\mathcal{X}}(x_1, x_2)}{\rho} \right)^2 \right) \right\}. \tag{5.3}$$

$g_{\mathcal{I}}$ can be used to directly replace $d^g_{\mathcal{I}}$ in (5.2). A small value of ρ focuses the energy on a local neighbourhood around the vertex, while a large value increases the receptive field. Using Gaussian kernels in W places more emphasis on local geometry whereas geodesics have higher values far away from the source vertex. Thus, geodesics work well for global alignment and Gaussians for local fine-tuning.

5.4.3 Cyclic α-Expansion (CAE)

CCuantuMM represents matchings as permutation matrices. In order to update them, we build on Seelbach *et al.*'s CAE algorithm [238] (similar to a fusion move [132]), which we describe here. A permutation matrix P is called an *r-cycle*, if there exist r disjoint indices $i_1, \ldots, i_r$ such that $P_{i_j i_{(j+1)\%r}} = 1$ for all $j \in \{1, \ldots, r\}$ and $P_{l,l} = 1$ for all $l \notin \{i_1, \ldots, i_r\}$, in which case $P = (i_1 i_2 \ldots i_r)$ is a common notation. Two cycles, *i.e.* two permutation matrices, are *disjoint* if these indices are pairwise disjoint. We know that disjoint cycles commute, which allows us to

represent any permutation P as $P = \left(\prod_{i=1}^{k} c_i\right)\left(\prod_{i=1}^{l} \tilde{c}_i\right)$, where $\{c_i\}_i$ and $\{\tilde{c}_i\}_i$ each are sets of disjoint 2-cycles.

Given a set $\{c_i\}_{i=1}^{k}$ of k disjoint 2-cycles, an *update*, or modification, of P can therefore be parameterised as: $P(\alpha) = \prod_{i=1}^{k} c_i^{\alpha_i} P$, where $\alpha \in \{0,1\}^k$ is a binary decision vector determining the update. (Note that α_i in $c_i^{\alpha_i}$ is an exponent, not an index.) Crucially, to make this parameterisation compatible with QUBOs, we need to make it linear in α. To this end, CAE uses the following equality:

$$P(\alpha) = P + \sum_{i=1}^{k} \alpha_i (c_i - I)P. \tag{5.4}$$

5.5 Our CCuantuMM Method

Previous adiabatic quantum computing methods [237, 238] can only match two shapes. We present a method for matching N shapes. To ensure cycle consistency on N shapes, it is sufficient that all triplets of shapes are matched cycle consistently [130]. CCuantuMM iteratively solves three-shape problems, which preserve cycle consistency by construction and fit on existing quantum annealers with limited resources. We introduce our formulation for matching three shapes in Sec. 5.5.1 and then extend it to N shapes in Sec. 5.5.2.

5.5.1 Matching Three Shapes

Consider the problem of matching three non-rigidly deformed shapes $\mathcal{S} = \{\mathcal{X}, \mathcal{Y}, \mathcal{Z}\}$ of n vertices each, while preserving cycle consistency. We formulate this as an energy minimisation with respect to $\mathcal{P} = \{P_{\mathcal{I}\mathcal{J}} \in \mathbb{P}_n | \mathcal{I}, \mathcal{J} \in \mathcal{S}\}$, the set of permutations between all pairs in $\mathcal{S}$:

$$\begin{aligned} \min_{\mathcal{P}} \quad & \sum_{\mathcal{I},\mathcal{J} \in \mathcal{S}; \mathcal{I} \neq \mathcal{J}} \operatorname{vec}(P_{\mathcal{I}\mathcal{J}})^\top W_{\mathcal{I}\mathcal{J}} \operatorname{vec}(P_{\mathcal{I}\mathcal{J}}), \\ \text{s.t.} \quad & P_{\mathcal{X}\mathcal{Z}} = P_{\mathcal{X}\mathcal{Y}} P_{\mathcal{Y}\mathcal{Z}}, \end{aligned} \tag{5.5}$$

where $W_{\mathcal{I}\mathcal{J}} \in \mathbb{R}^{n^2 \times n^2}$ is the energy matrix describing how well certain pairwise properties are conserved between shapes $\mathcal{I}$ and $\mathcal{J}$ (see Sec. 5.4.2), and $P_{\mathcal{X}\mathcal{Z}} = P_{\mathcal{X}\mathcal{Y}} P_{\mathcal{Y}\mathcal{Z}}$ enforces cyclic consistency. An overview of the algorithm for three shapes is shown in Alg. 3.

QUBO Derivation

To perform optimisation on the quantum annealer, we need to transform (5.5) into a QUBO problem. We adapt the CAE formulation from [238] (see Sec. 5.4.3) and iteratively update the permutations to decrease the value of (5.5). Given a set $C = \{c_i\}_{i=1}^{k}$ of k disjoint 2-cycles and binary decision variables α, we can parameterise our permutation matrices as $P_{\mathcal{I}\mathcal{J}}(\alpha) = P_{\mathcal{I}\mathcal{J}} + \sum_{i=1}^{k} \alpha_i (c_i - I) P_{\mathcal{I}\mathcal{J}}$. However, CAE alone is not sufficient to transform (5.5) into a QUBO as cyclic consistency is still missing. A simple solution would be to encourage cyclic consistency as a quadratic soft penalty, but then there are no guarantees on the solution. Instead, we enforce

cyclic consistency by construction:

$$P_{\mathcal{XZ}}(\alpha,\beta) = (P_{\mathcal{XY}} + \sum_{i=1}^{k} \alpha_i (c_i - I) P_{\mathcal{XY}}) \cdot (P_{\mathcal{YZ}} + \sum_{j=1}^{k} \beta_j (\tilde{c}_j - I) P_{\mathcal{YZ}}), \tag{5.6}$$

where $\{c_i\}_i$ ($\{\tilde{c}_j\}_j$) are cycles and α (β) are decision variables for the updates to $P_{\mathcal{XY}}$ ($P_{\mathcal{YZ}}$). For brevity, we write $C_i = (c_i - I)P_{\mathcal{XY}}$ and $\tilde{C}_j = (\tilde{c}_j - I)P_{\mathcal{YZ}}$. We explain how we construct the cycles in Secs. 5.5.1-5.5.1. Thus, we iteratively solve (5.5) via a sequence of problems of the form:

$$\min_{\alpha,\beta\in\{0,1\}^k} E_{\mathcal{XY}}(P_{\mathcal{XY}}(\alpha)) + E_{\mathcal{YZ}}(P_{\mathcal{YZ}}(\beta)) + E_{\mathcal{XZ}}(P_{\mathcal{XZ}}(\alpha,\beta)), \tag{5.7}$$

where $E_{\mathcal{IJ}}(P,Q) = \mathrm{vec}(P)^\top W_{\mathcal{IJ}}\,\mathrm{vec}(Q)$ and $E_{\mathcal{IJ}}(P) = E_{\mathcal{IJ}}(P,P)$. While the first two terms are in QUBO form, the third term contains cubic and bi-quadratic terms (see the supplement for details) which are not compatible with current quantum annealer architectures.

Higher-Order Terms. All of these higher-order terms come from $P_{\mathcal{XZ}}(\alpha,\beta)$, specifically from the term $H = \sum_i \sum_j \alpha_i\beta_j C_i\tilde{C}_j$. As we only consider 2-cycles, C_i and $\tilde{C}_j$ each have only four non-zero elements. Due to this extreme sparsity, most summands of H become 0.

We could tackle these undesirable terms by decomposing them into quadratic terms by using ancilla variables and adding penalty terms [83]. This gives exact solutions for sufficiently high weights of the penalty terms. However, multiple reasons speak against this: (1) the QUBO matrix is already dense (a clique) under the current formulation (as we will see in (5.9)) and adding ancilla qubits scales quadratically in k, (2) adding penalties makes the problem harder to solve, and (3) H is sparsely non-zero and in practise we observe no drastic influence on the quality of the solution.

Alternatively, we could assume H=0. However, this is unnecessarily strong since (1) H also contributes to quadratic terms ($E(H,\cdot)$), and (2) higher-order terms operating on the same decision variable trivially reduce to quadratic terms: $\alpha_i \cdot \alpha_i = \alpha_i$ for binary α_i. We thus keep those two types of terms and merely assume all truly cubic and bi-quadratic terms to be zero.

Cycle-Consistent CAE. After eliminating the higher-order terms and ignoring constants from (5.7), we obtain (with the same colour coding):

$$\begin{aligned}
\min_{\alpha,\beta} \quad & \sum_{i=1}^{k} \alpha_i \Big(F_{\mathcal{XY}}(P_{\mathcal{XY}}, C_i) + F_{\mathcal{XZ}}(P_{\mathcal{XZ}}, C_i P_{\mathcal{YZ}}) \Big) \\
& + \sum_{j=1}^{k} \beta_j \Big(F_{\mathcal{YZ}}(P_{\mathcal{YZ}}, \tilde{C}_j) + F_{\mathcal{XZ}}(P_{\mathcal{XZ}}, P_{\mathcal{XY}} \tilde{C}_j) \Big) \\
& + \sum_{i=1}^{k} \sum_{l=1}^{k} \alpha_i \alpha_l \Big(E_{\mathcal{XY}}(C_i, C_l) + E_{\mathcal{XZ}}(C_i P_{\mathcal{YZ}}, C_l P_{\mathcal{YZ}}) \Big) \\
& + \sum_{j=1}^{k} \sum_{l=1}^{k} \beta_j \beta_l \Big(E_{\mathcal{YZ}}(\tilde{C}_j, \tilde{C}_l) + E_{\mathcal{XZ}}(P_{\mathcal{XY}} \tilde{C}_j, P_{\mathcal{XY}} \tilde{C}_l) \Big) \\
& + \sum_{i=1}^{k} \sum_{j=1}^{k} \alpha_i \beta_j \Big(F_{\mathcal{XZ}}(P_{\mathcal{XY}} \tilde{C}_j, C_i P_{\mathcal{YZ}}) + F_{\mathcal{XZ}}(K_{ij}, P_{\mathcal{XZ}}) \\
& + F_{\mathcal{XZ}}(K_{ij}, P_{\mathcal{XY}} \tilde{C}_j) + F_{\mathcal{XZ}}(K_{ij}, C_i P_{\mathcal{YZ}}) + E_{\mathcal{XZ}}(K_{ij}, K_{ij}) \Big),
\end{aligned} \tag{5.8}$$

where $P_{\mathcal{XZ}} = P_{\mathcal{XZ}}(\mathbf{0}, \mathbf{0}) = P_{\mathcal{XY}} P_{\mathcal{YZ}}$, and we use the shorthands $F_{\mathcal{IJ}}(A, B) = E_{\mathcal{IJ}}(A, B) + E_{\mathcal{IJ}}(B, A)$ and $K_{ij} = C_i \tilde{C}_j$. Denoting $\alpha_{k+j} = \beta_j$ for an expanded $\alpha \in \{0, 1\}^{2k}$, (5.8) can be written in the form:

$$\min_{\alpha \in \{0,1\}^{2k}} \quad \alpha^\top \tilde{W} \alpha. \tag{5.9}$$

The full formula for $\tilde{W}$ is provided in the supplement. (5.9) is finally in QUBO form and we can optimise it classically or on real quantum hardware (see Sec. 5.4.1).

Choosing Vertices

The question of how to choose the sets of cycles $\{c_i\}_i, \{\tilde{c}_j\}_j$ is still open. We first choose a subset of vertices using the "worst vertices" criterion introduced in [238] based on the relative inconsistency $I_{\mathcal{XY}}$ of a vertex $x \in \mathcal{X}$ under the current permutation:

$$I_{\mathcal{XY}}(x) = \sum_{w \in \mathcal{X}} W_{x \cdot n + x^\top P_{\mathcal{XY}}, w \cdot n + w^\top P_{\mathcal{XY}}}, \tag{5.10}$$

where we treat the one-hot vector $x^\top P_{\mathcal{XY}}$ as a vertex index on $\mathcal{Y}$. A high value indicates that x is inconsistent with many other matches under $P_{\mathcal{XY}}$ and swapping it will likely improve the matching. We denote the set of the $m{=}2k$ vertices with the highest $I_{\mathcal{XY}}(\cdot)$ as $V_{\mathcal{X}}$. Finally, we follow the permutations to get $V_{\mathcal{Y}} = \{x^\top P_{\mathcal{XY}} | x \in V_{\mathcal{X}}\} \subset \mathcal{Y}$.

In practice, we observe a systematic improvement in the matchings when considering all three possibilities (using $I_{\mathcal{XY}}$, $I_{\mathcal{YZ}}$, or $I_{\mathcal{XZ}}$ as the starting point). We thus use three "sub"-iterations per iteration, one for each possibility.

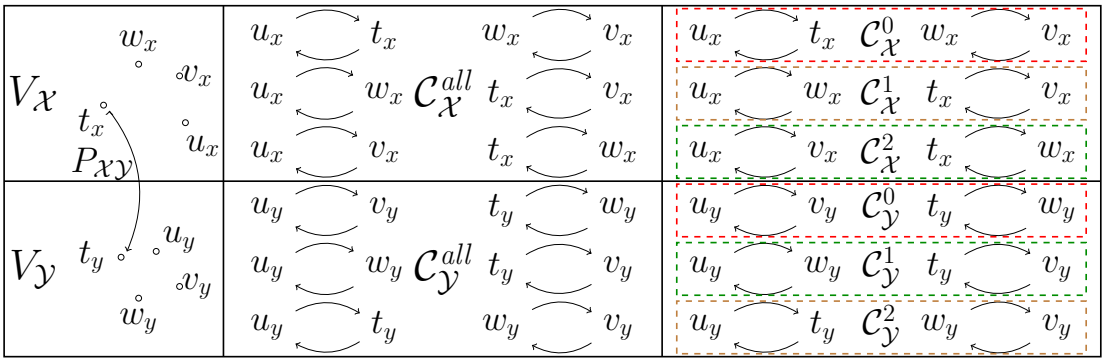


Figure 5.3: We depict the sub-iteration that starts from $I_{\mathcal{XY}}$, from which we construct $V_\mathcal{X}$, then $\mathcal{C}_\mathcal{X}^{all}$, and finally $\mathcal{C}_\mathcal{X} = \{\mathcal{C}_\mathcal{X}^0, \mathcal{C}_\mathcal{X}^1, \mathcal{C}_\mathcal{X}^2\}$. We also build $V_\mathcal{Y}$ from $V_\mathcal{X}$ and construct $\mathcal{C}_\mathcal{Y}$ analogously. Matching each element of $\mathcal{C}_\mathcal{X}$ with one from $\mathcal{C}_\mathcal{Y}$ (visualised via matching colours) leads to three sub-sub-iterations.

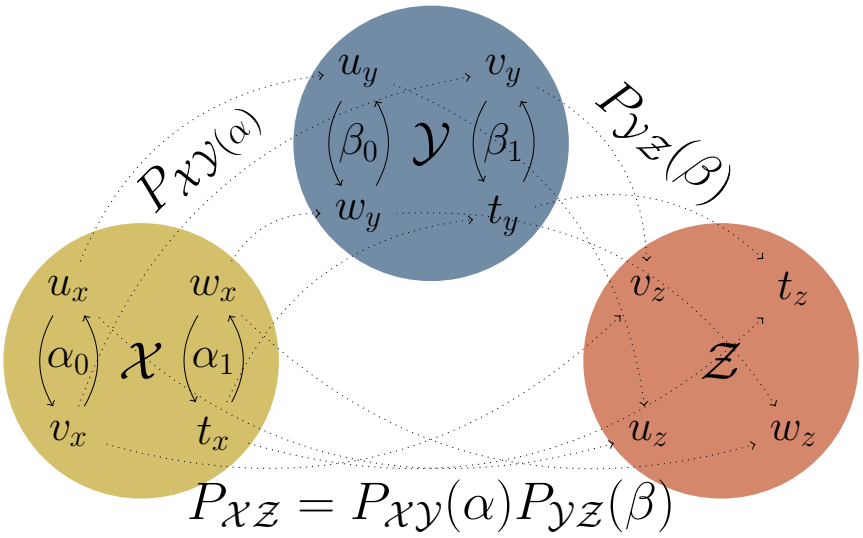


Figure 5.4: We depict the sub-sub-iteration for $\mathcal{C}_\mathcal{X}^2 = \{(u_x, v_x), (w_x, t_x)\}$ and $\mathcal{C}_\mathcal{Y}^1 = \{(u_y, w_y), (v_y, t_y)\}$ from Fig. 5.3.

Choosing Cycles

Given the worst vertices $V_{\mathcal{X}}$ and $V_{\mathcal{Y}}$ of any sub-iteration, we construct the cycles $\{c_i\}_i, \{\tilde{c}_j\}_j$ from them. Fig. 5.3 visualises this process. Focusing on $V_{\mathcal{X}}$ for the moment, we want to use all possible 2-cycles $\mathcal{C}_{\mathcal{X}}^{all} = \{(uv)|u, v \in V_{\mathcal{X}}, u \neq v\}$ in each sub-iteration. We cannot use all of these cycles at once since they are not disjoint, as CAE requires. Instead, we next construct a set $\mathcal{C}_{\mathcal{X}}$ by partitioning $\mathcal{C}_{\mathcal{X}}^{all}$ into $m-1$ sets of cycles with each containing $m/2{=}k$ disjoint cycles. An analogous methodology is used for $\mathcal{C}_{\mathcal{Y}}$.

We now have $\mathcal{C}_{\mathcal{X}}$ and $\mathcal{C}_{\mathcal{Y}}$. Since we want to consider each cycle of $\mathcal{C}_{\mathcal{X}}^{all}$ and $\mathcal{C}_{\mathcal{Y}}^{all}$ once, we need several "sub-sub" iterations. Thus, we next need to pick one set of cycles from each $\mathcal{C}_{\mathcal{X}}$ and $\mathcal{C}_{\mathcal{Y}}$ for each sub-sub-iteration. There are $(m-1)^2$ possible pairs between elements of $\mathcal{C}_{\mathcal{X}}$ and $\mathcal{C}_{\mathcal{Y}}$. Considering all possible pairs is redundant, does not provide significant performance advantage, and increases the computational complexity quadratically. Hence, we randomly pair each element of $\mathcal{C}_{\mathcal{X}}$ with one element of $\mathcal{C}_{\mathcal{Y}}$ (without replacement). This leads to $m-1$ sub-sub-iterations, with each one solving (5.9) with its respective cycles; see Fig. 5.4.

Algorithm 3 Hybrid Three-Shape Matching

Input: $\mathcal{P}^i, \mathcal{S}$
Output: $\mathcal{P}^{i+1}$

for $I \in \{I_{\mathcal{XY}}, I_{\mathcal{YZ}}, I_{\mathcal{XZ}}\}$ **do** ▷ sub-iterations
  construct $V_{\mathcal{X}}, V_{\mathcal{Y}}, V_{\mathcal{Z}}$ (see Sec. 5.5.1)
  construct $\mathcal{C}_{\mathcal{X}}, \mathcal{C}_{\mathcal{Y}}, \mathcal{C}_{\mathcal{Z}}$ (see Sec. 5.5.1)
  for $l{=}1$ to $m{-}1$ **do** ▷ sub-sub-iterations
    compute $\tilde{W}$
    optimise QUBO (5.9) ▷ quantum
  end for
  $P_{\mathcal{XY}} = \prod_{i=1}^{k} c_i^{\alpha_i} P_{\mathcal{XY}}$
  $P_{\mathcal{YZ}} = \prod_{j=1}^{k} \tilde{c}_j^{\alpha_{m+j}} P_{\mathcal{YZ}}$
  $P_{\mathcal{XZ}} = P_{\mathcal{XY}} \cdot P_{\mathcal{YZ}}$
end for
return $\mathcal{P}^{i+1} = \{P_{\mathcal{XY}}, P_{\mathcal{YZ}}, P_{\mathcal{XZ}}\}$

5.5.2 Matching N Shapes

In this section, we extend our model to matching a shape collection $\mathcal{S}$ with N elements by iteratively matching three shapes while still guaranteeing cycle consistency. Similar to the three-shape case (5.5), this can be formulated as an energy minimisation problem with respect to the set of permutations $\mathcal{P}$, except $\mathcal{S}$ now has cardinality N:

$$\begin{aligned} \min_{\mathcal{P}} \quad & \sum_{\mathcal{I},\mathcal{J} \in \mathcal{S}; \mathcal{I} \neq \mathcal{J}} E_{\mathcal{IJ}}(P_{\mathcal{IJ}}), \\ \text{s.t.} \quad & P_{\mathcal{IK}} = P_{\mathcal{IJ}} P_{\mathcal{JK}} \quad \forall \mathcal{I}, \mathcal{J}, \mathcal{K} \in \mathcal{S}. \end{aligned} \tag{5.11}$$

The energy contains summands for each possible pair of shapes. Solving all of them jointly would be computationally expensive and even more complicated than (5.8). This is the reason most multi-shape matching methods apply relaxations at this point or cannot scale to a large N.

However, the cycle-consistency constraints still only span over three shapes; triplets are sufficient for global consistency [130].

We thus iteratively focus on a triplet $\mathcal{X}, \mathcal{Y}, \mathcal{Z} \in \mathcal{S}$ and its set of permutations $\mathcal{P}' = \{P_{\mathcal{X}\mathcal{Y}}, P_{\mathcal{X}\mathcal{Z}}, P_{\mathcal{Y}\mathcal{Z}}\}$. We could then minimise (5.11) over $\mathcal{P}'$, leading to a block-coordinate descent optimisation of (5.11) over $\mathcal{P}$. This would make the problem tractable on current quantum hardware since it keeps the number of decision variables limited. It would also formally guarantee that our iterative optimisation would never increase the total energy. However, each iteration would be linear in N due to the construction of the QUBO matrix, preventing scaling to large N in practice. We therefore instead restrict (5.11) to those terms that depend *only* on permutations from $\mathcal{P}'$. This leads to the same energy as for the three-shape case (5.5), where the minimisation is now over $\mathcal{P}'$. Importantly, the computational complexity per triplet becomes independent of N, allowing to scale to large N. While this foregoes the formal guarantee that the *total* energy never increases, we crucially find that it still only rarely increases in practice; see the supplement.

By iterating over different triples $\mathcal{X}^i, \mathcal{Y}^i, \mathcal{Z}^i$, we cover the entire energy term and reduce it iteratively. Specifically, one iteration i of the N-shape algorithm runs Alg. 3 on $\mathcal{P}' = \{P^i_{\mathcal{X}^i\mathcal{Y}^i}, P^i_{\mathcal{Y}^i\mathcal{Z}^i}, P^i_{\mathcal{X}^i\mathcal{Z}^i}\}$. Here, $\mathcal{X}^i \in \mathcal{S}$ is chosen randomly (we use stratified sampling to pick all shapes equally often), the *anchor* $\mathcal{Y}^i = A \in \mathcal{S}$ is fixed, and $\mathcal{Z}^i = \mathcal{X}^{i-1}$. In practice, we saw slightly better results with this scheme instead of choosing the triplet randomly; see the supplement. We note that we only need to explicitly keep track of permutations into the anchor: $\mathcal{P}^i = \{P^i_{\mathcal{I}A}\}_{\mathcal{I}\in\mathcal{S},\mathcal{I}\neq A}$. We then get $\mathcal{P}^{i+1}$ from $\mathcal{P}^i$ by replacing $P^i_{\mathcal{X}^i A}$ and $P^i_{\mathcal{Z}^i A}$ with their updated versions from Alg. 3.

Initialisation. We compute an initial set of pairwise permutations $\mathcal{P}^{init}$ using a descriptor-based similarity of the normalised heat-kernel-signatures (HKS) [48] extended by a dimension indicating whether a vertex lies on the left or right side of a shape (a standard practice in the shape-matching literature [109]). Instead of using a random shape as anchor, the results improve when using the following shape:

$$A = \arg\min_{A\in\mathcal{S}} \sum_{\mathcal{I}\in\mathcal{S};\mathcal{I}\neq A} E_{\mathcal{I}A}(P^{init}_{\mathcal{I}A}), \tag{5.12}$$

where $P^{init}_{\mathcal{I}A} \in \mathcal{P}^{init}$. We thus have $\mathcal{P}^0 = \{P^{init}_{\mathcal{I}A}\}_{\mathcal{I}\in\mathcal{S},\mathcal{I}\neq A}$.

Time Complexity. Our algorithm scales linearly with the number of shapes. Each iteration of Alg. 3 has worst-case time complexity $\mathcal{O}(nk^3)$, as we discuss in the supplement.

Energy Matrix Schedule. In practise, we first use pure geodesics for a coarse matching and then Gaussian-filtered geodesics to fine-tune. Specifically, for a shape collection of three shapes, we use a schedule with $2T$ geodesics iterations followed by $2T$ Gaussian iterations. For each additional shape in the shape collection, we add T iterations to both schedules. We exponentially decrease the variance of the Gaussians every $N-1$ iterations to $\rho(i) = c_2 \exp\left(\frac{c_1}{i-T}\right)$ where c_1 and c_2 are chosen such that the variance decreases from 25% to 5% of the shape diameter over the iterations. Thus, *all* shapes undergo one iteration with the same specific variance. We refer to the supplement for more details.

5.6 Experimental Evaluation

We compare against state-of-the-art multi-matching methods with a focus on quantum methods. We consider classical works for reference. All experiments use Python 3.9 on an Intel Core i7-8565U CPU with 8GB RAM and the D-Wave Advantage System 4.1 (accessed via Leap 2). We will release our code, which is accelerated using Numba.

Hyperparameters. We set T=11. We set the number of worst vertices m to 16% of the number of vertices n.

Quantum Comparisons. The closest quantum work, Q-Match [238], matches only two shapes. We consider two adaptations to multi-matching: 1) Q-MatchV2-cc, similar to our CCuantuMM, chooses an anchor and matches the other shapes pairwise to it, implicitly enforcing cycle consistency; and 2) Q-MatchV2-nc matches all pairs of shapes directly, without guaranteed cycle consistency. In both cases, we use our faster implementation and adapt our energy matrix schedule, which gives significantly better results.

Classical Comparisons. For reference, we also compare against the classical, non-learning-based multi-matching state of the art: IsoMuSh [109] and the *synchronised* version of ZoomOut [186], which both guarantee vertex-wise cycle consistency across multiple shapes.

Evaluation Metric. We evaluate the correspondences using the Princeton benchmark protocol [144]. Given the ground-truth correspondences $P^*_{\mathcal{IJ}}$ for matching the shape $\mathcal{I}$ to $\mathcal{J}$, the error of vertex $v \in \mathcal{I}$ under our estimated matching $P_{\mathcal{IJ}}$ is given by the normalised geodesic distance:

$$e_v(P_{\mathcal{IJ}}) = \frac{d^g_{\mathcal{J}}(v^\top P_{\mathcal{IJ}}, v^\top P^*_{\mathcal{IJ}})}{\operatorname{diam}(\mathcal{J})}, \tag{5.13}$$

where $\operatorname{diam}(\cdot)$ is the shape diameter. We plot the fraction of errors that is below a threshold in a percentage-of-correct-keypoints (PCK) curve, where the threshold varies along the x-axis. As a summary metric, we also report the area-under-the-curve (AUC) of these PCK curves.

Datasets. The *FAUST* dataset [41] contains real scans of ten humans in different poses. We use the registration subset with ten poses for each class and downsample to 500 vertices. *TOSCA* [47] has 76 shapes from eight classes of humans and animals. We downsample to ∼1000 vertices. *SMAL* [280] has scans of toy animals in arbitrary poses, namely 41 non-isometric shapes from five classes registered to the same template. (*E.g.,* the felidae (cats) class contains scans of lions, cats, and tigers.) We downsample to 1000 vertices. We use the same number of vertices as IsoMuSh [109], except that they use 1000 vertices for FAUST.

5.6.1 Experiments on Real Quantum Annealer

We run two three-shapes and two ten-shapes experiments with FAUST on a real QPU. However, since our QUBO matrices are dense, we effectively need to embed a clique on the QPU. (The supplement contains a detailed analysis of the minor embeddings and the solution quality.) Hence, we test a reduced version of our method with 20 worst vertices per shape (40 virtual qubits in total), as more would worsen results significantly on current hardware. To compensate for this change, we use more iterations for the ten-shape experiments. We use 200 anneals per QUBO, the default annealing path, and the default annealing time of $20\mu s$. As standard chain strength, we choose 1.0001 times the largest absolute value of entries in Q. Each ten-shape experiment takes about 10 minutes of QPU time. In total, our results took about 30 minutes of QPU time for

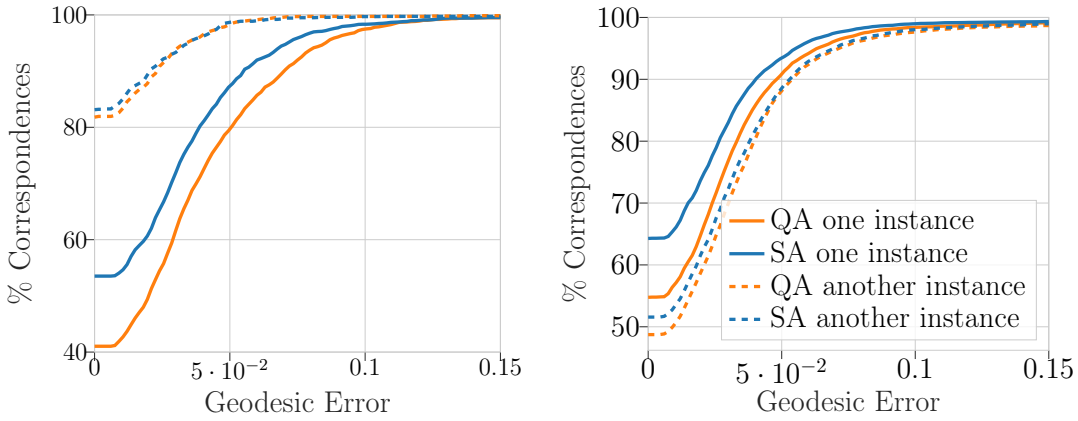


Figure 5.5: PCK curves for (left) two three-shape and (right) two ten-shape instances using QA and SA. In each plot, we denote one instance by normal lines and the other one by dotted lines.

a total of $5.5 \cdot 10^4$ QUBOs. QA under these settings achieves a similar performance as SA under the same settings (Fig. 5.5). As QPU time is expensive and since we have just shown that SA performs comparably to a QPU in terms of result quality, we perform the remaining experiments with SA under our default settings, on classical hardware. This is common practice [238, 18, 275] since SA is conceptually close to QA. For additional results, including results on the new Zephyr hardware [69], we refer to the supplement.

5.6.2 Comparison to Quantum and Classical SoTA

	Ours	Q-MatchV2-cc	Q-MatchV2-nc	IsoMuSh	ZoomOut	HKS
FAUST	**0.989**	0.886	0.879	*0.974*	0.886	0.746
TOSCA	**0.967**	0.932	0.940	*0.952*	0.864	0.742
SMAL	*0.866*	0.771	0.813	**0.926**	0.851	0.544

Table 5.1: AUC averaged over all classes of each dataset. For reference, we also include classical methods on the right.

FAUST. We outperform both quantum and classical prior work, as Fig. 5.7a and Tab. 5.1 show. Because we downsample FAUST more, IsoMuSh's results are better in our experiments than what Gao *et al.* [109] report.

Matching 100 Shapes. Next, we demonstrate that, unlike IsoMuSh and ZoomOut, our approach can scale to matching all 100 shapes of FAUST. Fig. 5.1 contains qualitative results. Tab. 5.2 compares the runtime of our method (using SA) to others. Only ours and Q-MatchV2-cc scale well to 100 shapes while ZoomOut and IsoMuSh cannot.

TOSCA. Fig. 5.7b and Tab. 5.1 show that our method achieves state-of-the-art results. While IsoMuSh's PCK curve starts higher (better), the AUC in Tab. 5.1 suggests that our method performs better overall. Fig. 5.6 has qualitative examples.

SMAL. Our CCuantuMM outperforms the quantum baselines, both in terms of PCK (Fig. 5.7c)

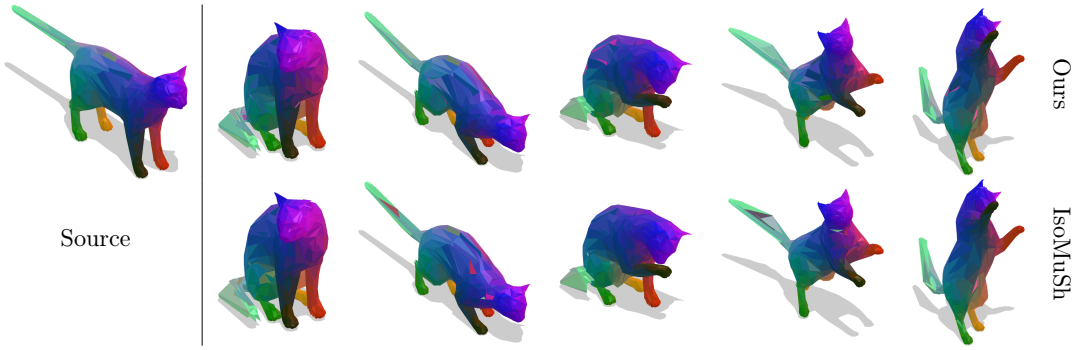


Figure 5.6: Qualitative results on the TOSCA [47] cat class. We colour a source shape and transfer this colouring to target shapes via the matches estimated by our method or IsoMuSh [109].

# Shapes	Ours	Q-MatchV2-cc	Q-MatchV2-nc	IsoMuSh	ZoomOut
10	97	16	81	(4+)0.3	**4**
100	1137	**175**	∼8000†	OOM	OOM

Table 5.2: Runtime (in *min*) for FAUST. IsoMuSh uses ZoomOut for initialisation. "OOM" (out of memory): memory requirements are infeasible. "†" denotes an estimate.

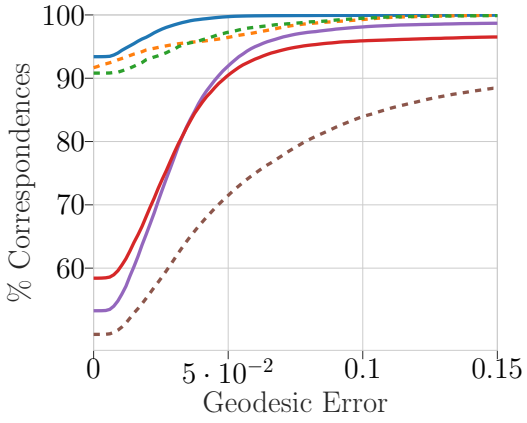
and AUC (Tab. 5.1). At the same time, it achieves performance on par with ZoomOut and below IsoMuSh. SMAL is considered the most difficult of the three datasets due to the challenging non-isometric deformations of its shapes. All methods thus show worse performance compared to FAUST and TOSCA.

5.6.3 Ablation Studies

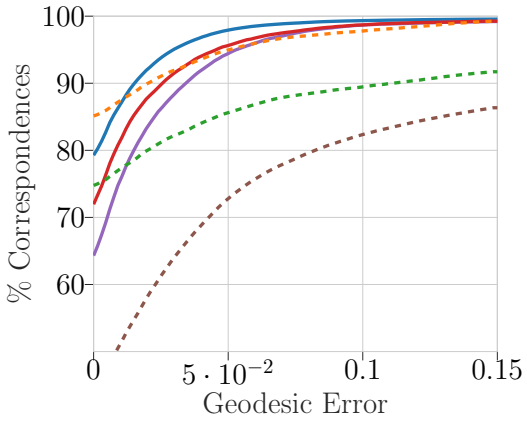
We perform an ablation study on FAUST to analyse how different components of our method affect the quality of the matchings. We refer to the supplement for more ablations.

Gaussian Energy Schedule. Our schedule, which starts with geodesics and afterwards uses Gaussians, provides a significant performance gain over using only geodesics, under the same number of iterations, see Fig 5.8. That is because Gaussians better correct local errors in our approach.

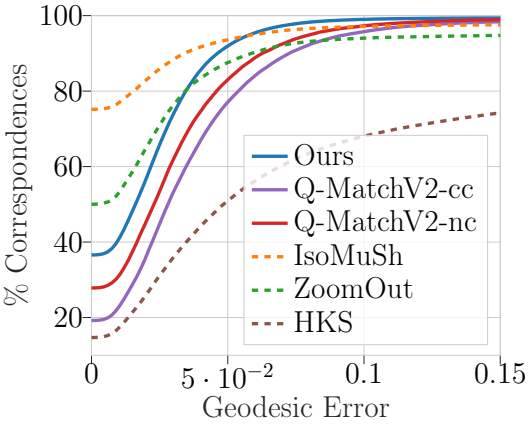
Does Using More Shapes Improve Results? We analyse what effect increasing the number of shapes N has on the matchings' quality. We first randomly select three shapes and run our method on them, to obtain the baseline. Next, we run our method again and again from scratch, each time adding one more shape to the previously used shapes. This isolates the effect of using more shapes from all other factors. In Fig. 5.8, we plot the PCK curves *for the three selected shapes*. We repeat this experiment for several randomly sampled instances. Our results show that including more shapes improves the matchings noticeably overall.



(a) FAUST [41]



(b) TOSCA [47]



(c) SMAL [280]

Figure 5.7: Quantitative results on all three datasets. For each dataset, we match all shapes within a class and then plot the average PCK curve across classes. We plot classical methods with dashed lines as they are only for reference. HKS is our initialisation (see Sec. 5.5.2).

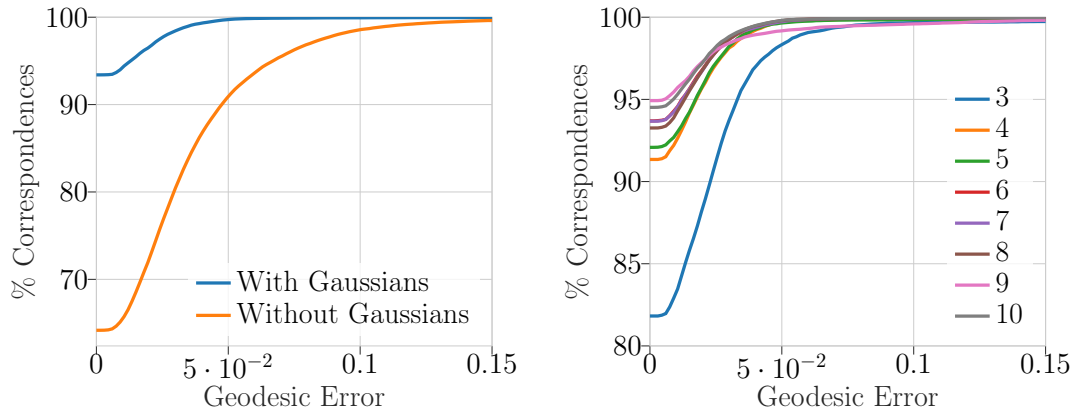


Figure 5.8: We ablate (left) the usage of Gaussian kernels, and (right) the large-scale multi-shape setting. Gaussian kernels improve the results greatly. Matching more shapes improves results.

5.6.4 Discussion and Limitations

Our method and all considered methods are based on intrinsic properties like geodesic distances. Thus, without left-right labels for initialisation, they would produce partial flips for inter-class instances in FAUST and intra-class instances in TOSCA and SMAL. For a large worst-vertices set, contemporary quantum hardware leads to embeddings (see Sec. 5.4.1) with long chains, which are unstable, degrading the result quality. Finally, while our method is currently slower in practice than SA, it would immediately benefit from the widely expected quantum advantage in the future.

5.7 Conclusion

The proposed method achieves our main goal: improving mesh alignment w.r.t. the quantum state of the art. Furthermore, it is even highly competitive among classical state-of-the-art methods. This suggests that the proposed approach can be used as a reference for comparisons and extensions of classical mesh-alignment works in the future. (For such cases, classical SA is a viable alternative when access to quantum computers is lacking.) Our results show that ignoring certain higher-order terms still allows for high-quality matchings, which is promising for future quantum approaches that could use similar approximations. Finally, unlike classical work, we designed our method within the constraints of contemporary quantum hardware. We found that iteratively considering shape triplets is highly effective, perhaps even for classical methods.

Acknowledgements. This work was partially supported by the ERC Consolidator Grant 4DReply (770784). ZL is funded by the Ministry of Culture and Science of the State of NRW.

5.8 Supplementary Material

This supplementary material contains additional results. We present them in the order they are mentioned in the main paper. First, Sec. 5.9 provides an overview of our notation. Sec. 5.10

formally states the N-shape algorithm. Sec. 5.11 contains large-scale figures showing the 100 matched FAUST shapes. Sec. 5.12 describes the elimination of higher-order terms in detail. Sec. 5.13 states the weight matrix $\tilde{W}$ of the final QUBO. Sec. 5.14 shows that the total energy almost never decreases in practice. Sec. 5.15 compares our anchor scheme to using random triplets. Sec. 5.16 discusses the time complexity of our proposed method. Sec. 5.17 provides more implementation details on initialisation and deciding trivial cycles. Sec. 5.18 discusses the minor embeddings on real quantum hardware and contains further QPU experiments. Sec. 5.19 presents more results, including more ablation experiments. Finally, Sec. 5.20 compares existing quantum computer vision works for alignment tasks.

In addition, the supplementary material contains a video showing the evolution of the matchings of a ten-shape FAUST instance over the course of the optimisation. We visualise the matchings by fixing the colouring of the top-left shape and transferring this colouring to all other shapes according to the estimated correspondences.

5.9 Notations

Tab. 5.3 summarises the notation we use.

5.10 N-Shape Algorithm

We provide the formal N-shape algorithm as Alg. 4.

5.11 Matching 100 Shapes

Fig. 5.9 shows qualitative results of all 100 matched FAUST shapes.

5.12 Higher-Order Terms

In this section, we give the expansions for the third term of (7), namely $E_{\mathcal{XZ}}(P_{\mathcal{XZ}}(\alpha, \beta))$, as it contains higher-order terms. Fig. 5.20 shows the equations. The terms in red are cubic or bi-quadratic and we hence assume them to be 0, except for the summands that simplify to quadratic terms. For example, $\alpha_i\beta_j\alpha_i\beta_j E_{\mathcal{XZ}}(C_i\tilde{C}_j, C_i\tilde{C}_j)$ simplifies to the linear summand $\alpha_i\beta_j E_{\mathcal{XZ}}(C_i\tilde{C}_j, C_i\tilde{C}_j)$. This then yields (8) from the main paper.

5.13 Final QUBO

Fig. 5.10 shows the weight matrix $\tilde{W} \in \mathbb{R}^{2k\times 2k}$ that is used for the final QUBO.

5.14 Evolution During Optimisation

Fig. 5.11 depicts how the total energy (11) and PCK evolve during optimisation. Importantly, the energy almost never increases in practice. Furthermore, our algorithm converges close to the

Algorithm 4 CCuantuMM (Matching N Shapes; Sec. 4.2)

Input: $\mathcal{S}, T$
Output: $\mathcal{P}$

initialise $\mathcal{P}^{init}$ from HKS descriptors
determine anchor $A \in \mathcal{S}$
$\mathcal{P}^0 \leftarrow \{P^{init}_{\mathcal{I}A}\}_{\mathcal{I}\in\mathcal{S},\mathcal{I}\neq A}$
randomly pick $\mathcal{X}^{-1} \in \mathcal{S}$ ▷ technicality for $\mathcal{Z}^0$
for $i = 0$ to $2T(|\mathcal{S}| - 1)$ **do** ▷ iterations
  if $i \mod (|\mathcal{S}| - 1) == 0$ **then**
    $\mathcal{S}' \leftarrow \mathcal{S} \setminus \{A\}$ ▷ for stratified sampling
  end if
  randomly pick $\mathcal{X}^i \in \mathcal{S}'$
  $\mathcal{S}' \leftarrow \mathcal{S}' \setminus \{\mathcal{X}^i\}$
  $\mathcal{Y}^i \leftarrow A$
  $\mathcal{Z}^i \leftarrow \mathcal{X}^{i-1}$
  get $P^i_{\mathcal{X}^iA}, P^i_{\mathcal{Z}^iA}$ from $\mathcal{P}^i$
  $P^i_{\mathcal{X}^i\mathcal{Y}^i} \leftarrow P^i_{\mathcal{X}^iA}$
  $P^i_{\mathcal{Y}^i\mathcal{Z}^i} \leftarrow (P^i_{\mathcal{Z}^iA})^{-1}$
  $P^i_{\mathcal{X}^i\mathcal{Z}^i} \leftarrow P^i_{\mathcal{X}^iA}(P^i_{\mathcal{Z}^iA})^{-1}$
  mode $\leftarrow$ geodesic **if** $i < T(|\mathcal{S}| - 1)$ **else** Gaussian
  $\{P^{i+1}_{\mathcal{X}^i\mathcal{Y}^i}, P^{i+1}_{\mathcal{Y}^i\mathcal{Z}^i}, P^{i+1}_{\mathcal{X}^i\mathcal{Z}^i}\} \leftarrow$ run Alg. 1 with mode on $(\{P^i_{\mathcal{X}^i\mathcal{Y}^i}, P^i_{\mathcal{Y}^i\mathcal{Z}^i}, P^i_{\mathcal{X}^i\mathcal{Z}^i}\}, \{\mathcal{X}^i, \mathcal{Y}^i, \mathcal{Z}^i\})$
  $P^{i+1}_{\mathcal{X}^iA} \leftarrow P^{i+1}_{\mathcal{X}^i\mathcal{Y}^i}$
  $P^{i+1}_{\mathcal{Z}^iA} \leftarrow (P^{i+1}_{\mathcal{Y}^i\mathcal{Z}^i})^{-1}$
  $\mathcal{P}^{i+1} \leftarrow (\mathcal{P}^i \setminus \{P^i_{\mathcal{X}^iA}, P^i_{\mathcal{Z}^iA}\}) \cup \{P^{i+1}_{\mathcal{X}^iA}, P^{i+1}_{\mathcal{Z}^iA}\}$
end for
return $\mathcal{P} = \mathcal{P}^{2T(|\mathcal{S}|-1)}$

ground truth. A sudden and significant improvement occurs as soon as the schedule switches from geodesics to Gaussians. We note that although the objective uses Gaussians in the second half of the schedule, both the total energy plotted here and the PCK are based on the (unfiltered) geodesic distances.

5.15 Fixed Anchor vs. Random Triplets

Fig. 5.12 compares our anchor-based scheme with using random triplets. We see a small improvement with our scheme.

5.16 Time Complexity

Our method mainly consists of constructing and solving the QUBO matrix $\tilde{W}$, which is based on W. However, standard meshes contain thousands of vertices, which makes naïvely calculating the full $W \in \mathbb{R}^{n^2\times n^2}$ not feasible due to the memory restrictions. Fortunately, we do not need to compute the full W but only a small set of its entries. This is due to the extreme sparsity of the $c_i - I$ matrix (only four non-zero elements) since we only consider 2-cycles. Using k 2-cycles

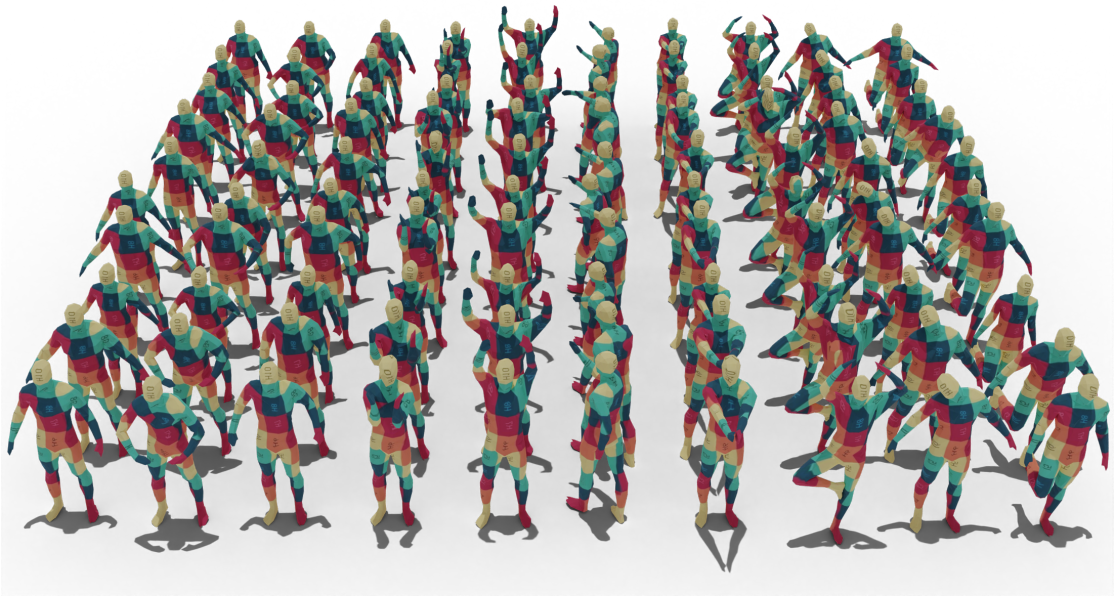
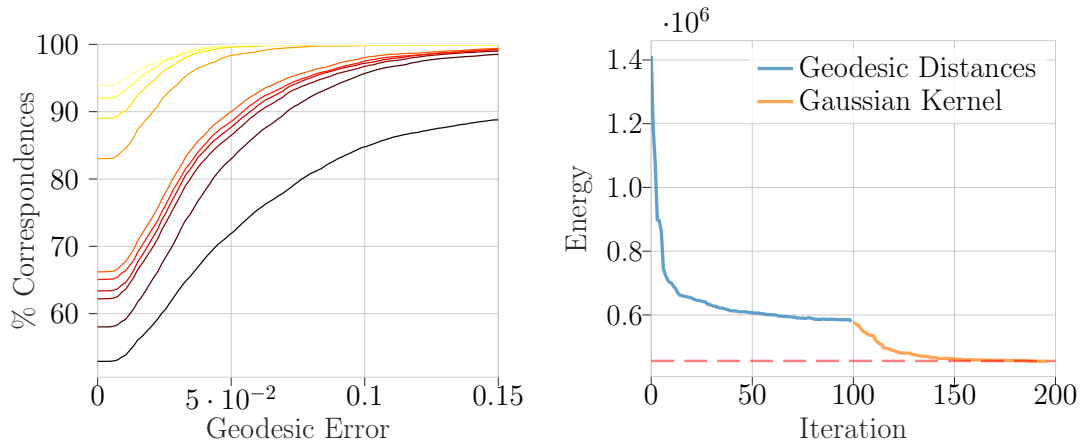


Figure 5.9: **Extended version of Fig. 1 in the main paper.** We visualise the matchings between all 100 FAUST shapes via texture transfer. All correspondences are cycle-consistent by design.

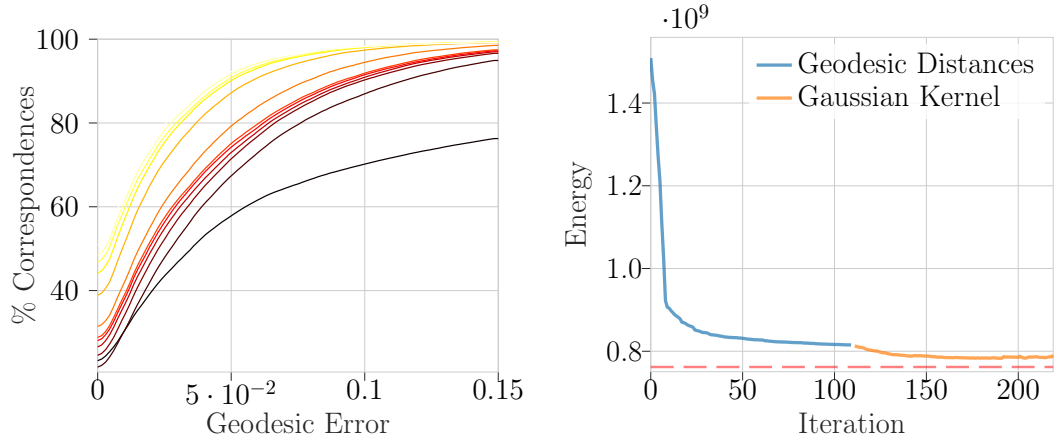
leads to a worst-case time complexity of $\mathcal{O}(nk^2)$ for the three-shape Alg. 1 from the main paper (*i.e.*, one sub-sub-iteration). For the sub-sub-iterations of a sub-iteration, $\tilde{W}$ is constant and we thus need to compute $\tilde{W}$ only once for each sub-iteration. In addition, each iteration has $k-1$ sub-iterations, resulting in a time complexity of $\mathcal{O}(nk^3)$ for each iteration. Furthermore, our implementation for computing $\tilde{W}$ is significantly faster in practice than the sub-sampling technique proposed in Q-Match.

$$\tilde{W}_{ij} = \begin{cases} E_{\mathcal{YZ}}(C_i, C_i) + E_{\mathcal{XY}}(P_{\mathcal{XY}}, C_i) + E_{\mathcal{XY}}(C_i, P_{\mathcal{XY}}) + E_{\mathcal{XZ}}(P_{\mathcal{XY}}P_{\mathcal{YZ}}, C_iP_{\mathcal{YZ}}) + E_{\mathcal{XZ}}(C_iP_{\mathcal{YZ}}, P_{\mathcal{XY}}P_{\mathcal{YZ}}) + E_{\mathcal{XZ}}(C_iP_{\mathcal{YZ}}, C_iP_{\mathcal{YZ}}) & \text{if } i = j \leq k \\ E_{\mathcal{XY}}(\tilde{C}_j, \tilde{C}_j) + E_{\mathcal{YZ}}(P_{\mathcal{YZ}}, \tilde{C}_j) + E_{\mathcal{YZ}}(\tilde{C}_j, P_{\mathcal{YZ}}) + E_{\mathcal{XZ}}(P_{\mathcal{XY}}P_{\mathcal{YZ}}, P_{\mathcal{XY}}\tilde{C}_j) + E_{\mathcal{XZ}}(P_{\mathcal{XY}}\tilde{C}_j, P_{\mathcal{XY}}P_{\mathcal{YZ}}) + E_{\mathcal{XZ}}(P_{\mathcal{XY}}\tilde{C}_j, P_{\mathcal{XY}}\tilde{C}_j) & \text{if } i = j > k \\ E_{\mathcal{YZ}}(C_i, C_j) + E_{\mathcal{XZ}}(C_iP_{\mathcal{YZ}}, C_jP_{\mathcal{YZ}}) & \text{if } i \neq j, i \leq k, j \leq k \\ E_{\mathcal{XY}}(\tilde{C}_i, \tilde{C}_j) + E_{\mathcal{XZ}}(P_{\mathcal{XY}}\tilde{C}_i, P_{\mathcal{XY}}\tilde{C}_j) & \text{if } i \neq j, i > k, j > k \\ E_{\mathcal{XZ}}(P_{\mathcal{XY}}P_{\mathcal{YZ}}, C_i\tilde{C}_j) + E_{\mathcal{XZ}}(P_{\mathcal{XY}}\tilde{C}_j, C_iP_{\mathcal{YZ}}) + E_{\mathcal{XZ}}(P_{\mathcal{XY}}\tilde{C}_j, C_i\tilde{C}_j) + E_{\mathcal{XZ}}(C_iP_{\mathcal{YZ}}, P_{\mathcal{XY}}\tilde{C}_j) & \\ + E_{\mathcal{XZ}}(C_iP_{\mathcal{YZ}}, C_i\tilde{C}_j) + E_{\mathcal{XZ}}(C_i\tilde{C}_j, P_{\mathcal{XY}}P_{\mathcal{YZ}}) + E_{\mathcal{XZ}}(C_i\tilde{C}_j, P_{\mathcal{XY}}\tilde{C}_j) & \\ + E_{\mathcal{XZ}}(C_i\tilde{C}_j, C_iP_{\mathcal{YZ}}) + E_{\mathcal{XZ}}(C_i\tilde{C}_j, C_i\tilde{C}_j) & \text{if } i \neq j, i \leq k, j > k \\ 0 & \text{otherwise} \end{cases} \tag{5.14}$$

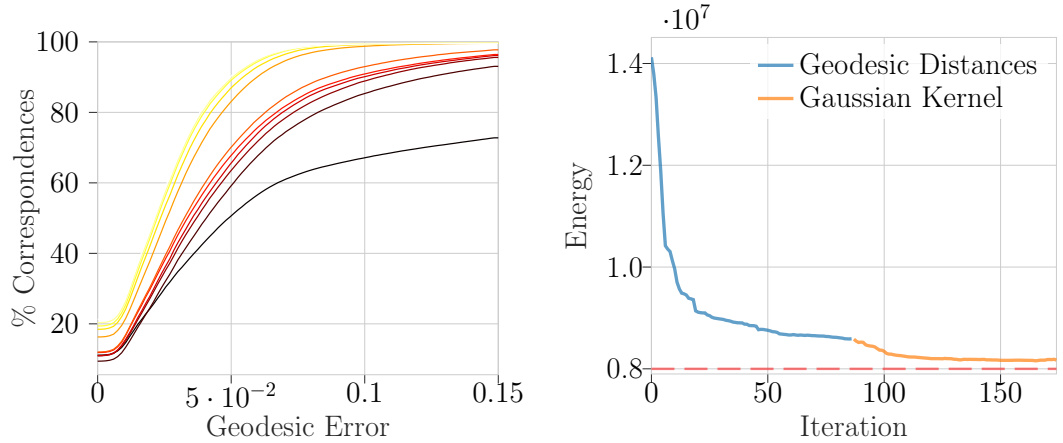
Figure 5.10: The final QUBO weight matrix $\tilde{W}$. For indices $i > k$, we define $\tilde{C}_i = \tilde{C}_{i-k}$.



(a) FAUST



(b) TOSCA



(c) SMAL

Figure 5.11: Evolution during the optimisation (left) of the PCK, depicted with a colour bar gradient, and (right) of the total energy. The horizontal red-dashed line is the energy of the ground-truth solution. The results are on (a) a class of FAUST containing ten shapes, (b) the cat class of TOSCA containing eleven shapes, and (c) the cat class of SMAL containing nine shapes.

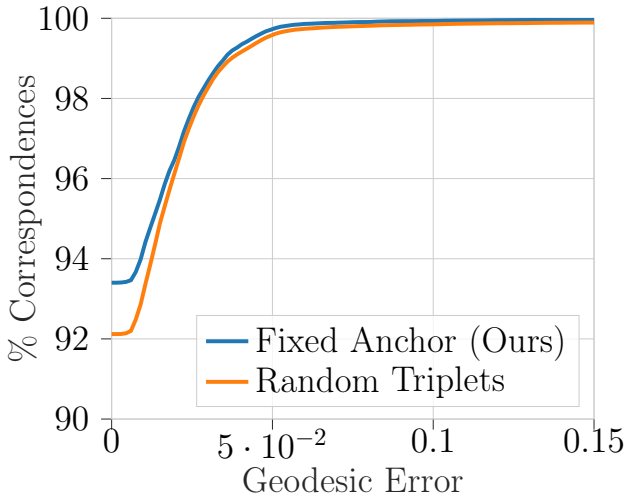


Figure 5.12: PCK when using a fixed anchor (as our method does) and when using random triplets. We average across all classes of FAUST.

5.17 Implementation Details

5.17.1 Initialisation

The initial set of permutations $\mathcal{P}^{init}$ is computed using a descriptor-based similarity $DS_{\mathcal{IJ}} \in \mathbb{R}^{n \times n}$ between all n vertices of shape pairs $\mathcal{I}, \mathcal{J} \in \mathcal{S}$. Specifically, $DS_{\mathcal{IJ}}(u, v)$ contains the similarity (inner product) of the normalised heat-kernel-signatures (HKS) [48] descriptors (which we extend by an additional dimension indicating whether a vertex lies on the left or right side of a shape) of vertex u of $\mathcal{I}$ and vertex v of $\mathcal{J}$. The left-right descriptors reduce left-right flips in the solution since the shape classes we consider are globally symmetric, which is not captured by the local HKS descriptors. This is standard practice in the shape-matching literature [109]. The solution of a linear assignment problem on $DS_{\mathcal{IJ}}$ is then the initial $P^{init}_{\mathcal{IJ}}$.

5.17.2 Pre-Computing Trivial Cycles

The matrix $\tilde{W}$ consists of couplings (quadratic terms) and linear terms. Numerical experiments show that there often exist cycles with linear terms that dominate the corresponding coupling terms. This happens when a cycle is largely uncorrelated to the rest of the cycles in the current set of cycles. In this case, the decision for such a cycle can be made trivially.

We now derive an inequality for how large the linear term has to be such that the couplings can be neglected in the optimisation problem. Consider the QUBO problem:

$$\min_{\alpha \in \{0,1\}^k} \alpha^T W \alpha + b^T \alpha, \tag{5.15}$$

where α are the decision variables, $b \in \mathbb{R}^k$ is a vector, and W is a symmetric matrix with zeros on the diagonal representing the couplings. We can look at the terms that depend on α_q for fixed

q separately:

$$\alpha^T W \alpha = \alpha_q \sum_{i \neq q} W_{q,i}\alpha_i + \alpha_q \sum_{i \neq q} \alpha_i W_{i,q} + \sum_{j \neq q}(\sum_{i \neq q} \alpha_j W_{j,i}\alpha_i + \alpha_j b_j) + \alpha_q b_q.$$

As W is symmetric, $W_{i,q} = W_{q,i}$ holds and we can write

$$\alpha^T W \alpha = \alpha_q \left(b_q + 2 \sum_{i \neq q} W_{q,i}\alpha_i \right) + \sum_{j \neq q}(\sum_{i \neq q} \alpha_j W_{j,i}\alpha_i + \alpha_j b_j).$$

It follows that if:

$$|b_q| \geq \sum_{i \neq q} 2|W_{q,i}|, \tag{5.16}$$

then we can make the decision based on the sign of b_q: If b_q is positive we do not choose the cycle as doing so would increase the energy. This reduces the number of physical qubits required for the embedding.

5.18 Minor Embeddings and Other QPU Experiments

5.18.1 QPU Processing and Annealing Time

Optimising a single QUBO uses $\sim 40ms$ of total QPU processing time for 200 anneals. This is also called the *QPU access time* [68]. However, there are also several overheads that occur when solving QUBOs by accessing a D-Wave annealer via the cloud. For example, a latency when connecting to the D-Wave annealer and a post processing time. The QPU access time also includes a programming time.

These overheads can be orders of magnitude greater than the time taken by the actual annealing, which is very short since we use the default annealing schedule and the default annealing time of 20 μs.

5.18.2 Minor Embeddings

As explained in the main paper, not all physical qubits on a real quantum processing unit (QPU) can be connected (coupled) with each other. Thus, a minor embedding of the logical-qubit graph (defined by non-zero entries of the QUBO matrix) into the physical-qubit graph (defined by the hardware) is required. This can lead to a chain of multiple physical qubits representing a single logical qubit.

Our logical input graph is a clique. Due to the limited connectivity of current hardware, a clique cannot be directly embedded onto the physical annealer. We thus require a minor embedding, which is commonly computed using Cai *et al.*'s method [52]. Fig. 5.21 visualises

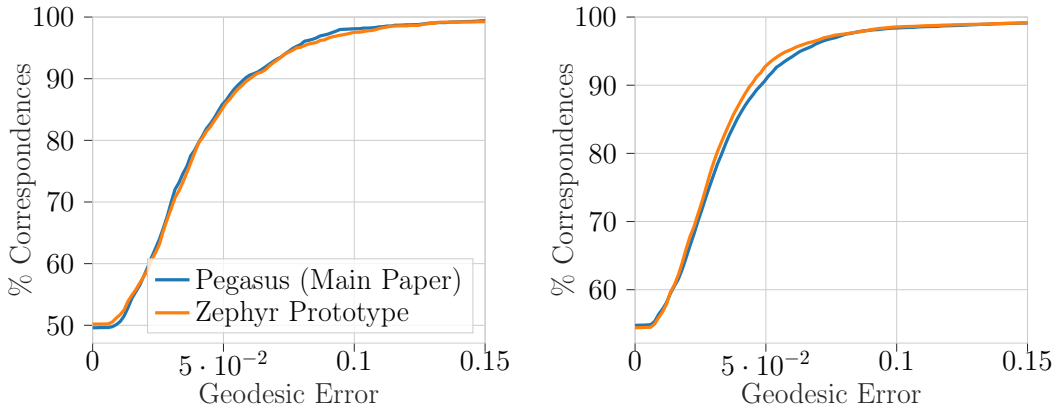


Figure 5.13: PCK curves for (left) a three-shape and (right) a ten-shape inter-class FAUST instance on both QPU architectures.

an example minor embedding. We note that, since our input graphs are cliques, a generalised embedding can be pre-computed and reused, not impacting the time complexity.

5.18.3 Minor Embeddings in Practice

In this section, we investigate the empirical impact of minor embeddings on the solutions. The minor embeddings depend on the qubit topology, *i.e.* the physical qubit connectivity pattern. Here, we show results on the D-Wave Advantage 4.1 with its Pegasus topology (used in the main paper) and also first results on a D-Wave Advantage2 prototype of the next-generation Zephyr topology, which has a higher connectivity than Pegasus.

Fig. 5.13 shows PCK curves on both topologies when using $m{=}2k{=}20$ worst vertices. Both architectures obtain similar results, although they are very slightly worse than SA. However, as discussed in the main paper, we find that the performance of CCuantuMM degrades significantly when using more than 20 worst vertices with QA. Specifically, Fig. 5.14 shows that the quality of the matchings worsens when using 40 worst vertices on both topologies, with Zephyr obtaining slightly better results. Still, in both cases, the quality is worse than the matching quality obtained by SA. Only SA shows the desired behaviour of improving when more worst vertices are used.

The cause for these results lies with the structure of the minor embeddings and not the plain number of physical qubits. Fig. 5.15 shows how the structural properties of the minor embeddings evolve as the number of worst vertices increases. For 20 worst vertices, they are similar. However, for 40 worst vertices, Zephyr uses fewer physical qubits and smaller chains, which explains the very slight performance advantage in Fig. 5.14. Physical qubits in a chain representing a single logical qubit are less likely to all anneal to the same value the longer the chain is. Longer chains become unstable and hence inconsistent, leading to inferior solution quality.

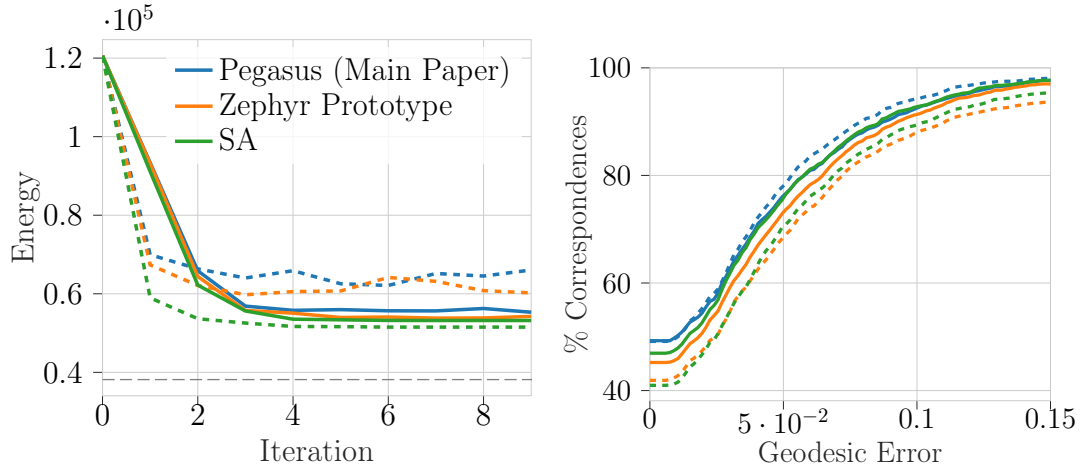


Figure 5.14: Quantitative results when using (solid) 20 and (dashed) 40 worst vertices on a three-shape FAUST instance. We show (left) the energy evolution during optimisation and (right) the final PCK curves.

5.19 Further Results and Ablations

5.19.1 Additional Qualitative Results

Fig. 5.22, Fig. 5.23, and Fig. 5.24 provide additional qualitative examples of matchings on TOSCA and SMAL calculated with our method and the competitors.

5.19.2 Variation Within a Dataset

Within a dataset, some instances have more difficult deformations and are inherently harder to match than easier instances, independent of the method employed. We investigate the extent of this variation by taking a closer look at TOSCA. We observe a significant variation of PCK curves across different classes in Fig. 5.16 and of AUC in Tab. 5.4.

5.19.3 Influence of Descriptors

We ablate the need for left-right indicators when using HKS descriptors for initialisation. Fig. 5.17 contains results on the cat class of TOSCA. Without left-right indicators, we observe flips in the matchings on TOSCA and partial flips on FAUST for inter-class instances. This is expected since both our method and IsoMuSh exploit intrinsic properties of the shapes, which are invariant to such symmetric flips.

5.19.4 Noise Perturbation

We investigate the robustness of our approach to noise. To that end, we analyse the effect of adding synthetic perturbations to the shapes. Specifically, for each FAUST mesh, we add a Gaussian-distributed offset along the vertex normal to each vertex position. This ensures the meshed structure of the shape does not change. Fig. 5.18 visualises the amount of noise we

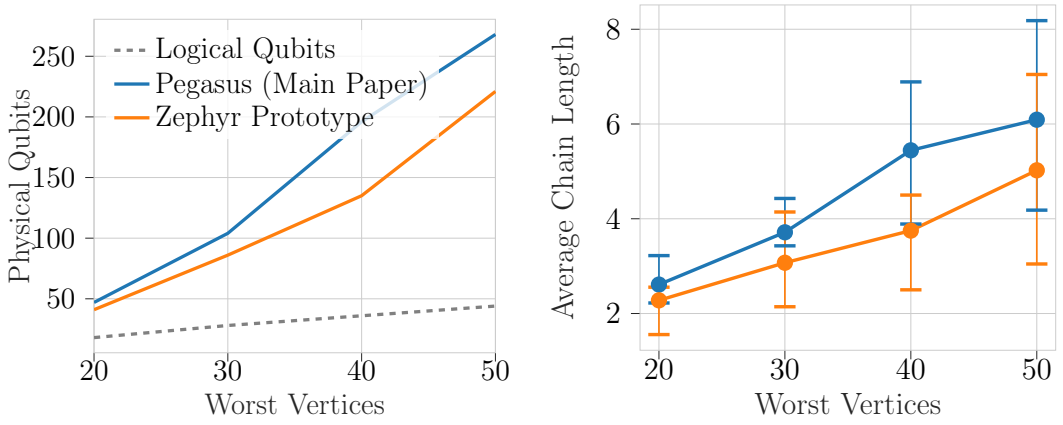


Figure 5.15: Structural changes of the minor embeddings when using more worst vertices. We show (left) the number of physical qubits and (right) the average chain length, for both QPU topologies. The number of *logical* qubits equals the number of worst vertices.

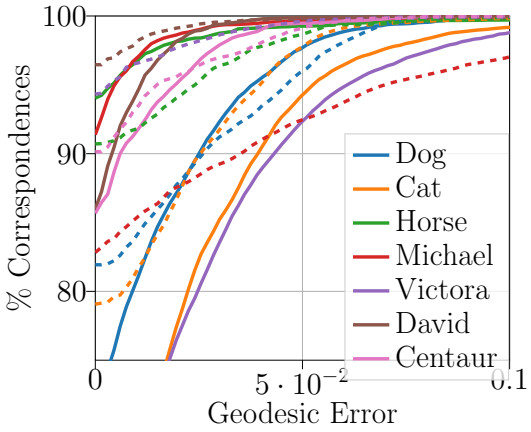


Figure 5.16: PCK for seven different classes of TOSCA. We plot our method with solid lines and IsoMuSh [109] with dashed lines.

experiment with. Fig. 5.19 shows that our method is significantly more robust to noise than IsoMuSh.

5.20 Related Quantum Computer Vision Works

Several quantum methods tackle alignment tasks, as Tab. 5.5 shows. However, only Q-Match is relevant for comparisons. Several works only consider two point clouds and cannot handle the multi-matching setting of our work, do not consider fully non-rigid transformations, or only operate on point clouds, not meshes. QGM [237] only considers two graphs with at most four vertices. Q-Sync [40] similarly only works on at most five vertices, far fewer than what our method can handle.

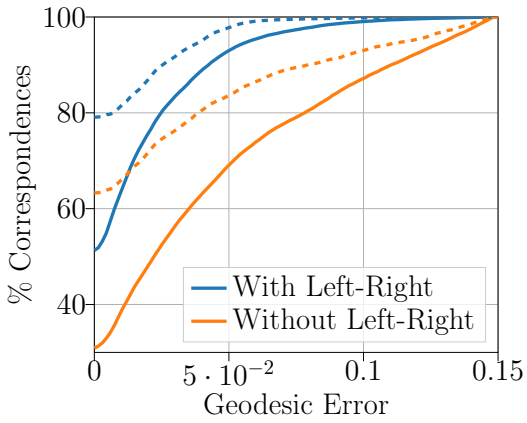


Figure 5.17: PCK with and without left-right descriptors. We plot our method with solid lines and IsoMuSh [109] with dashed lines.

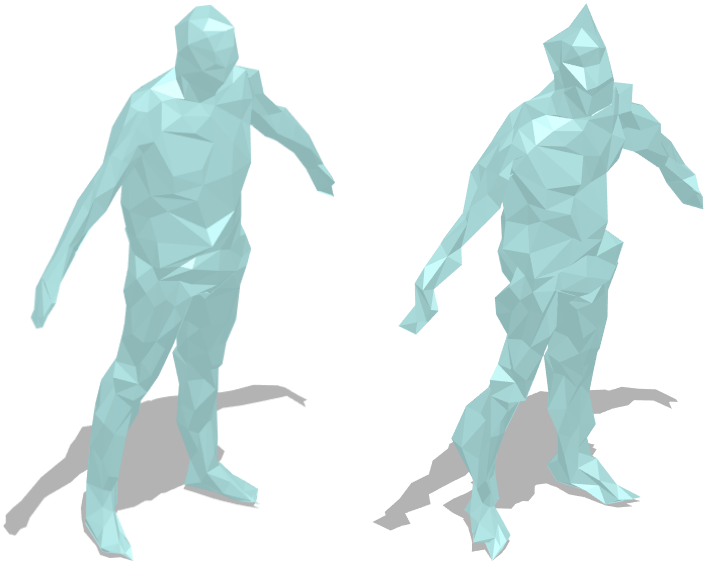


Figure 5.18: Example of perturbing the geometry on FAUST, with the noise variance set to 0.01 (left) and 0.02 (right).

Notation	Meaning
$\mathcal{I}$, $\mathcal{J}$, $\mathcal{X}$, $\mathcal{Y}$, $\mathcal{Z}$	shapes (represented as meshes)
$P_{\mathcal{IJ}}$	permutation matrix from shape $\mathcal{I}$ to $\mathcal{J}$
x, y, u, v	vertices of a shape
$\mathcal{P}$	set of permutations
$\mathcal{S}$	set of shapes
W	energy matrix
N	number of shapes
I	identity matrix
$E_{\mathcal{IJ}}(P,Q)$	$vec(P)^\top W_{\mathcal{IJ}}\, vec(Q)$
$E_{\mathcal{IJ}}(P)$	$E_{\mathcal{IJ}}(P,P)$
$F_{\mathcal{IJ}}(A,B)$	$E_{\mathcal{IJ}}(A,B) + E_{\mathcal{IJ}}(B,A)$
α, β	decision variables
k	size of decision variables α,β
m	number of worst vertices
$I_{\mathcal{XY}}(x)$	relative inconsistency of vertex x for $P_{\mathcal{XY}}$
$V_{\mathcal{X}}$	set of worst vertices for shape $\mathcal{X}$
$\mathcal{C}_{\mathcal{X}}$	set of cycles for shape $\mathcal{X}$
A	anchor shape
C_i	$(c_i - I)P_{\mathcal{XY}}$; single-cycle update of $P_{\mathcal{XY}}$
$\tilde{C}_j$	$(c_j - I)P_{\mathcal{YZ}}$; single-cycle update of $P_{\mathcal{YZ}}$
K_{ij}	$C_i\tilde{C}_j$

Table 5.3: Notation used in this work.

	Dog	Cat	Horse	Michael	Victoria	David	Centaur
Ours	0.957	0.917	**0.990**	**0.992**	0.912	0.989	0.983
IsoMush	**0.959**	**0.959**	0.981	0.939	**0.991**	**0.996**	**0.984**

Table 5.4: AUC across seven different classes of TOSCA.

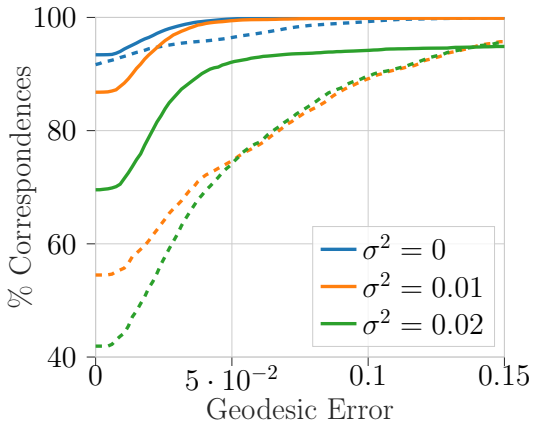


Figure 5.19: PCK for different amounts of perturbation (Gaussian noise with variance σ^2). We plot our method with solid lines and IsoMuSh [109] with dashed lines. We report results on a class of FAUST.

$$
\begin{aligned}
E_{\mathcal{XZ}}(P_{\mathcal{XZ}}(\alpha,\beta)) &= E_{\mathcal{XZ}}\left((P_{\mathcal{XY}}+\sum_{i=1}^{k}\alpha_i C_i)\cdot(P_{\mathcal{YZ}}+\sum_{j=1}^{k}\beta_j\tilde{C}_j),(P_{\mathcal{XY}}+\sum_{q=1}^{k}\alpha_q C_q)\cdot(P_{\mathcal{YZ}}+\sum_{l=1}^{k}\beta_l\tilde{C}_l)\right)\\
&= E_{\mathcal{XZ}}\left(P_{\mathcal{XY}}P_{\mathcal{YZ}},(P_{\mathcal{XY}}+\sum_{q=1}^{k}\alpha_q C_q)\cdot(P_{\mathcal{YZ}}+\sum_{l=1}^{k}\beta_l\tilde{C}_l\right)+E_{\mathcal{XZ}}\left(\sum_{j=1}^{k}\beta_j P_{\mathcal{XY}}\tilde{C}_j),(P_{\mathcal{XY}}+\sum_{q=1}^{k}\alpha_q C_q)\cdot(P_{\mathcal{YZ}}+\sum_{l=1}^{k}\beta_l\tilde{C}_l)\right)\\
&+E_{\mathcal{XZ}}\left(\sum_{i=1}^{k}\alpha_i C_i P_{\mathcal{YZ}},(P_{\mathcal{XY}}+\sum_{q=1}^{k}\alpha_q C_q)\cdot(P_{\mathcal{YZ}}+\sum_{l=1}^{k}\beta_l\tilde{C}_l)\right)+E_{\mathcal{XZ}}\left(\sum_{i=1}^{k}\sum_{j=1}^{k}\alpha_i\beta_j C_i\tilde{C}_j,(P_{\mathcal{XY}}+\sum_{q=1}^{k}\alpha_q C_q)\cdot(P_{\mathcal{YZ}}+\sum_{l=1}^{k}\beta_l\tilde{C}_l)\right)\\
&= E_{\mathcal{XZ}}(P_{\mathcal{XY}}P_{\mathcal{YZ}},P_{\mathcal{XY}}P_{\mathcal{YZ}})+\sum_{l=1}^{k}\beta_l E_{\mathcal{XZ}}(P_{\mathcal{XY}}P_{\mathcal{YZ}},P_{\mathcal{XY}}\tilde{C}_l)+\sum_{q=1}^{k}\alpha_q E_{\mathcal{XZ}}(P_{\mathcal{XY}}P_{\mathcal{YZ}},C_q P_{\mathcal{YZ}})\\
&+\sum_{q=1}^{k}\sum_{l=1}^{k}\alpha_q\beta_l E_{\mathcal{XZ}}(P_{\mathcal{XY}}P_{\mathcal{YZ}},C_q\tilde{C}_l)+\sum_{j=1}^{k}\beta_j E_{\mathcal{XZ}}(P_{\mathcal{XY}}\tilde{C}_j,P_{\mathcal{XY}}P_{\mathcal{YZ}})+\sum_{j=1}^{k}\sum_{l=1}^{k}\beta_j\beta_l E_{\mathcal{XZ}}(P_{\mathcal{XY}}\tilde{C}_j,P_{\mathcal{XY}}\tilde{C}_l)\\
&+\sum_{j=1}^{k}\sum_{q=1}^{k}\beta_j\alpha_q E_{\mathcal{XZ}}(P_{\mathcal{XY}}\tilde{C}_j,C_q P_{\mathcal{YZ}})+\sum_{j=1}^{k}\sum_{q=1}^{k}\sum_{l=1}^{k}\beta_j\alpha_q\beta_l E_{\mathcal{XZ}}(P_{\mathcal{XY}}\tilde{C}_j,C_q\tilde{C}_l)+\sum_{i=1}^{k}\alpha_i E_{\mathcal{XZ}}(C_i P_{\mathcal{YZ}},P_{\mathcal{XY}}P_{\mathcal{YZ}})\\
&+\sum_{i=1}^{k}\sum_{l=1}^{k}\alpha_i\beta_l E_{\mathcal{XZ}}(C_i P_{\mathcal{YZ}},P_{\mathcal{XY}}\tilde{C}_l)+\sum_{i=1}^{k}\sum_{q=1}^{k}\alpha_i\alpha_q E_{\mathcal{XZ}}(C_i P_{\mathcal{YZ}},C_q P_{\mathcal{YZ}})+\sum_{i=1}^{k}\sum_{q=1}^{k}\sum_{l=1}^{k}\alpha_i\alpha_q\beta_l E_{\mathcal{XZ}}(C_i P_{\mathcal{YZ}},C_q\tilde{C}_l)\\
&+\sum_{i=1}^{k}\sum_{j=1}^{k}\alpha_i\beta_j E_{\mathcal{XZ}}(C_i\tilde{C}_j,P_{\mathcal{XY}}P_{\mathcal{YZ}})+\sum_{i=1}^{k}\sum_{j=1}^{k}\sum_{l=1}^{k}\alpha_i\beta_j\beta_l E_{\mathcal{XZ}}(C_i\tilde{C}_j,P_{\mathcal{XY}}\tilde{C}_l)+\sum_{i=1}^{k}\sum_{j=1}^{k}\sum_{q=1}^{k}\alpha_i\beta_j\alpha_q E_{\mathcal{XZ}}(C_i\tilde{C}_j,C_q P_{\mathcal{YZ}})\\
&+\sum_{i=1}^{k}\sum_{j=1}^{k}\sum_{q=1}^{k}\sum_{l=1}^{k}\alpha_i\beta_j\alpha_q\beta_l E_{\mathcal{XZ}}(C_i\tilde{C}_j,C_q\tilde{C}_l)
\end{aligned}
\tag{5.17}
$$

Figure 5.20: We expand the third term of (7) from the main paper, which yields higher-order terms (highlighted in red). When the summands constituting these terms are truly cubic or bi-quadratic, we assume them to be 0, which results in the QUBO (8).

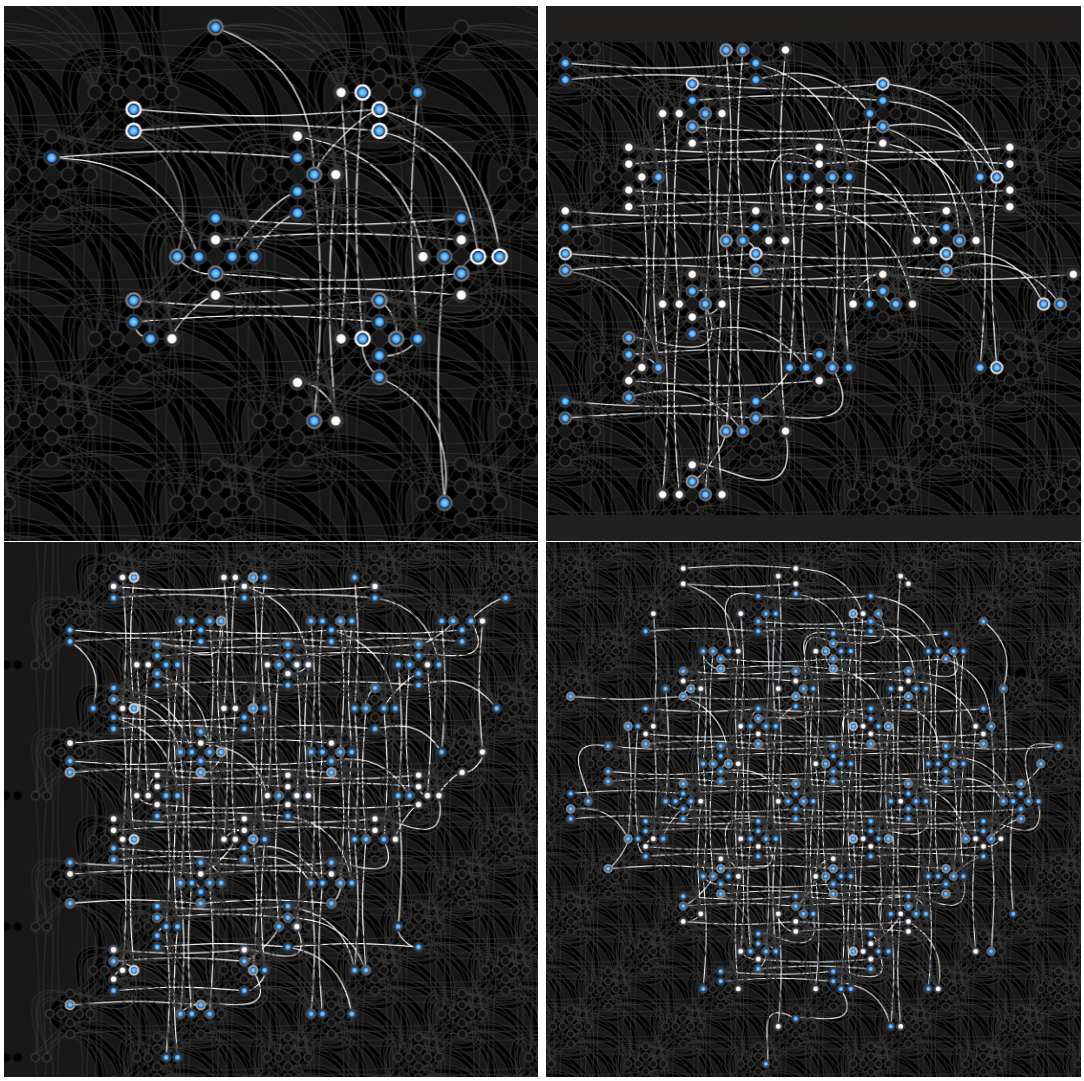


Figure 5.21: Visualisation of an example minor embedding on Pegasus (which we use in the main paper). The visualisation is obtained via D-Wave Leap 2's problem inspector [80] for (upper left) 20, (upper right) 30, (lower left) 40, and (lower right) 50 worst vertices. Each node depicts a physical qubit and the edges depict the chains of the minor embeddings.

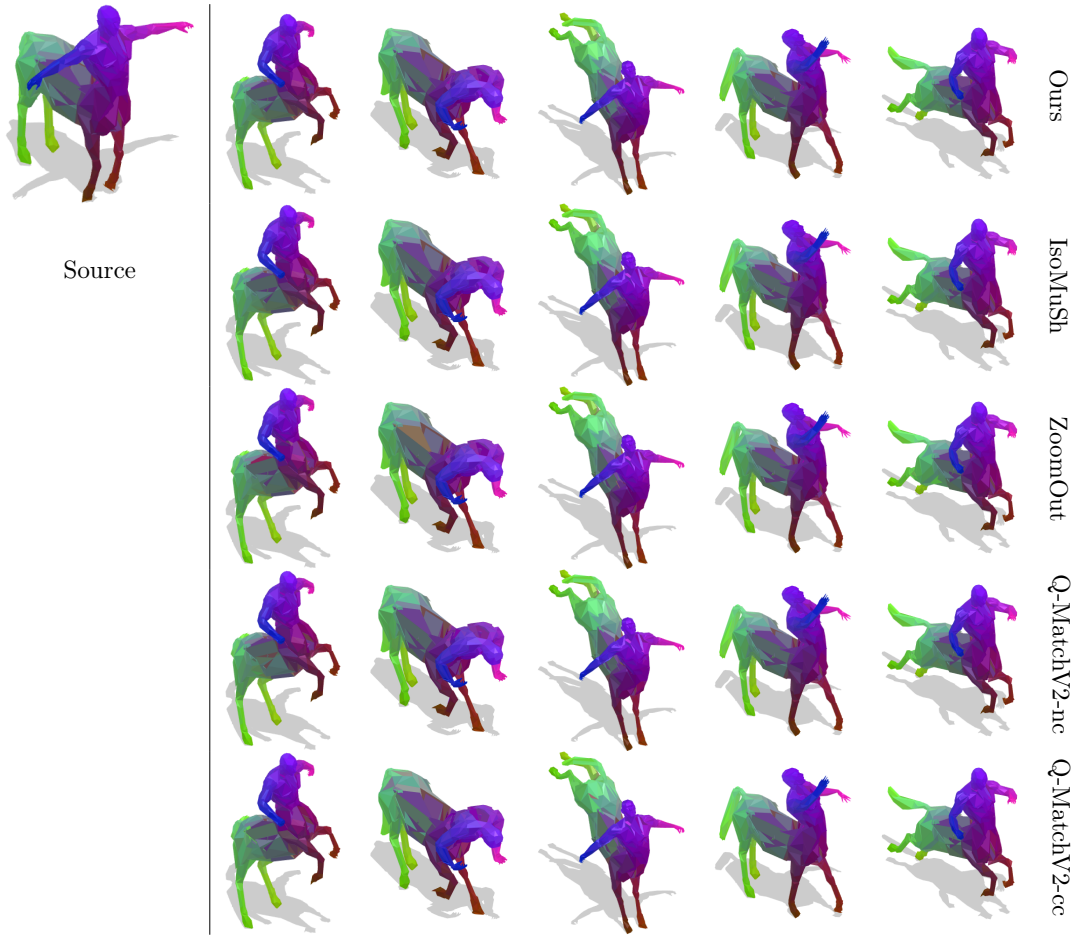


Figure 5.22: Qualitative results on the TOSCA [47] centaur class. We colour a source shape and transfer this colouring to target shapes via the matches estimated by our method and competitors.

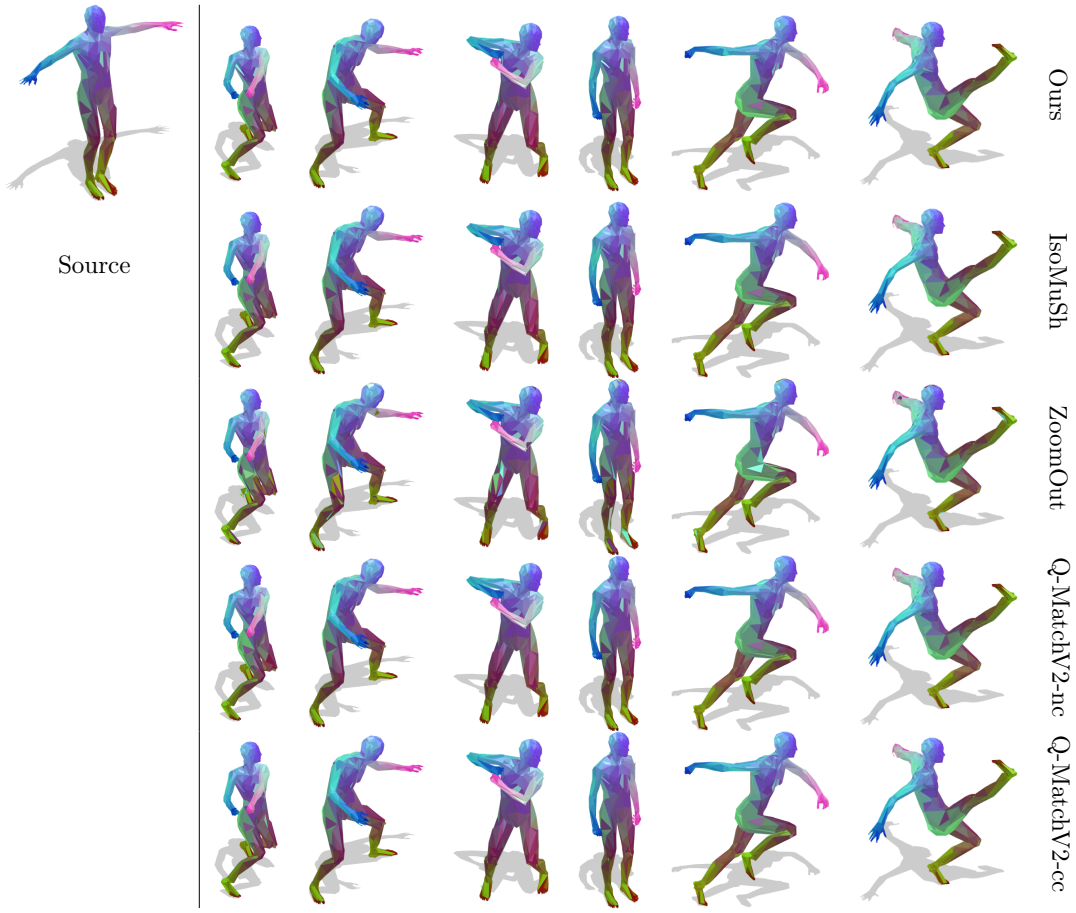


Figure 5.23: Qualitative results on the TOSCA [47] David class. We colour a source shape and transfer this colouring to target shapes via the matches estimated by our method and competitors.

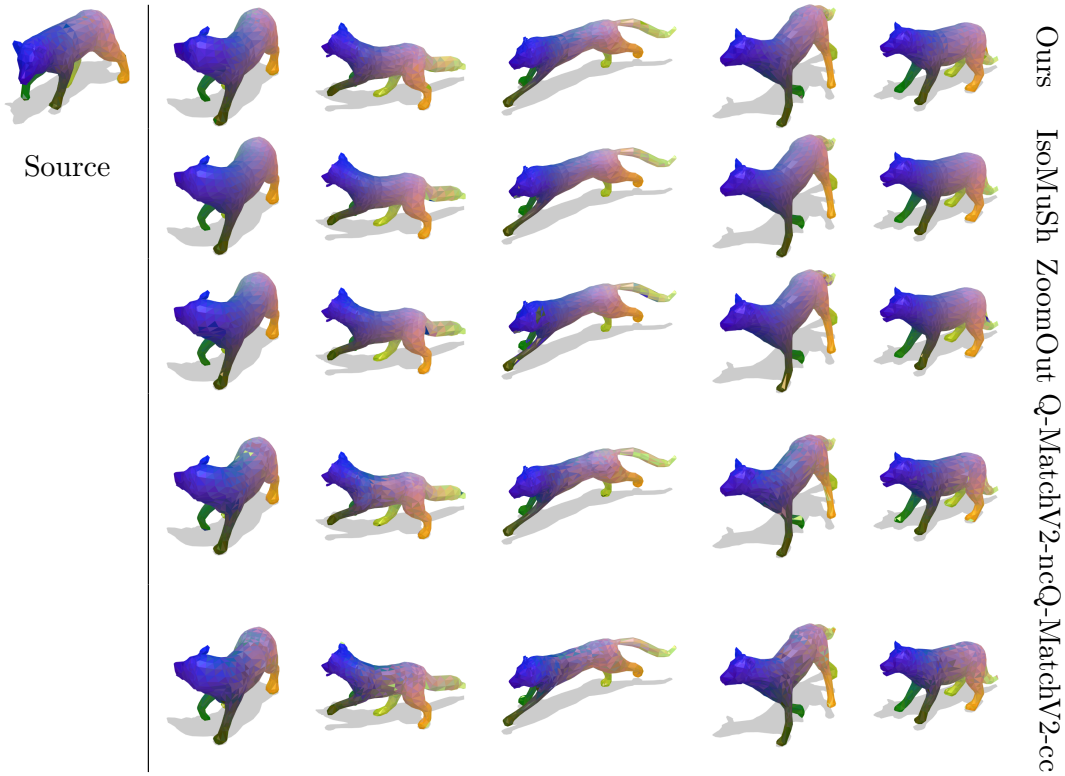


Figure 5.24: Qualitative results on a subset of the SMAL dog class. We colour a source shape and transfer this colouring to target shapes via the matches estimated by our method and competitors.

Method	Problem	Transformation	Input Type	# Inputs	# Points	# Qubits (per sweep)	QPU	Iterative
QA, CVPR 2020 [111]	TE, PSR	A/R	point clouds	2	$\leq 5k^\star$	$\leq$140	2000Q	
IQT, CVPR 2022 [185]	TE	R	point clouds	2	$\leq 1.5k^\star$	$\leq$10	2000Q	✓
qKC, ICASSP 2022 [198]	PSR	R	point clouds	2	$\leq 2k^\star$	4-6	Rigetti [65]	✓
QGM, 3DV 2020 [237]	GM	F	graphs	2	≤ 4	$\leq$50	2000Q	
QSync, CVPR 2021 [40]	PS, GM	F	perm. matrices	≤ 5	≤ 5	$\leq 1.5k$	Adv1.1	
QuMoSeg, ECCV 2022 [18]	MS	F	segm. matrices	≤ 9	$\leq$200	$\leq$ 250	Adv{1.1;4.1}	
Q-Match, ICCV 2021 [238]	MA	R/NR	meshes	2	$\leq$500	$\leq$200	Adv4.1	✓
CCuantuMM (Ours)	MA	R/NR	meshes	$\leq$100	$\leq 1k$	$\leq$50	Adv4.1, Adv2 prototype	✓

Table 5.5: Overview of related quantum methods for alignment tasks on point sets, graphs and meshes. "$\star$": according to the experiments reported in the paper; the methods can also process larger point clouds. Key: "TE": transformation estimation; "PSR": point set alignment; "GM": graph matching; "PS": permutation synchronisation; "MS": motion segmentation; "MA": mesh alignment; "A": affine transformation; "R": rigid transformation; "NR": non-rigid deformations; "F": the method operates on features extracted in a pre-processing step (and can support both rigid and non-rigid transformations). Note that only Q-Match [238] can be extended and applied to our data.

Chapter 6

QuAnt: Quantum Annealing with Learnt Couplings

6.1 Contextualization

The ICLR publication "Learning QUBO Forms in Quantum Annealing" is about integrating QUBO solvers in neural networks. From another perspective, one can also describe it as trying to learn the best QUBO formulation of a given computer vision problem. In previous publications, like QGM, we have already observed that there are various different ways to write a specific application problem as a QUBO. The goal here is now to not derive these formulations per hand every time, but instead to use training data.

Marcel Seelbach Benkner and Maximilian Krahn performed the experiments. Vladislav Golyanik had the idea of learning the coupling matrix of the QUBOs in the way it is done here and also influenced, together with Edith Tretschk the experiments via valuable, active discussion. Michael Möller and Zorah Lähner significantly contributed to writing, in the discussions and to designing the experiments.

6.2 Introduction

Hybrid computer vision methods that can be executed partially on a quantum computer (QC) are an emerging research area [45, 58, 238, 274]. Compared to classical methods, they promise to solve computationally demanding (*e.g.,* combinatorial) sub-problems faster, with improved scaling, and without relaxations that often lead to approximate solutions. Although quantum primacy has not yet been demonstrated in remotely practical usages of quantum computing, all existing quantum computer vision (QCV) methods fundamentally assume that it will be achieved in the future. Thus, solving these suitable algorithmic parts on a QC has the potential to reshape the field. However, reformulating them for execution on a QC is often non-trivial.

QCV continues building up momentum, fuelled by accessible experimental quantum annealers (QA) allowing to solve practical ($\mathcal{NP}$-hard) optimisation problems. Existing QCV methods using QAs rely on analytically deriving QUBOs (both QUBO matrices and solution encodings) for a specific problem type, which is challenging, especially since solutions need to be encoded as

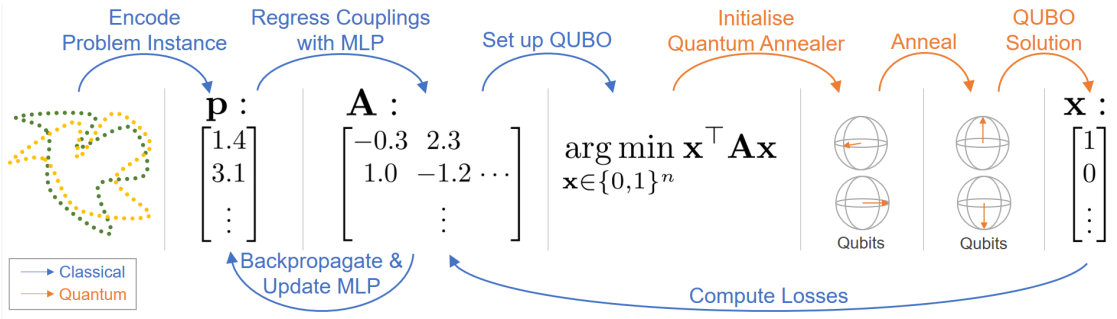


Figure 6.1: We propose **QuAnt for QUBO learning**, *i.e.,* a quantum-classical meta-learning algorithm that avoids analytical QUBO derivations by learning to regress QUBOs to solve problems of a given type. We first represent a problem instance as a vector $\mathbf{p}$ and then feed it into an MLP that regresses the entries of the QUBO matrix $\mathbf{A}$. We then initialise a quantum annealer with $\mathbf{A}$ and use quantum annealing to find a QUBO minimiser and extract it as the solution $\mathbf{x}^*$ to the problem instance. We define losses involving $\mathbf{x}^*$ that avoid backpropagation through the annealing and backpropagate gradients through the MLP to train it. We demonstrate the generalisability of QuAnt on graph matching, point set registration, and rotation estimation.

binary vectors [163, 237, 238, 40]. This often leads to larger encodings than necessary, severely impacting scalability. Alternatively, QUBO derivations with neural networks are conceivable but have not yet been scrutinised in the QA literature.

In stark contrast to the state of the art, this paper proposes, for the first time, to learn QUBO forms from data for any problem type using backpropagation (see Fig. 6.1). Our framework captures, in the weights of a neural network, the entire subset of QUBOs belonging to a problem type; a single forward pass yields the QUBO form for a given problem instance. It is thus a meta-learning approach in the context of hybrid (quantum-classical) neural network training, in which the superordinate network instantiates the parameters of the QUBO form. We find that sampling instantiated QUBOs can be a reasonable alternative to non-quantum neural baselines that regress the solution directly.

In particular, we show how a (combinatorial) quantum annealing solver can be integrated into a vanilla neural network as a custom layer and be used in the forward and backward passes, which may be useful in other contexts. To that end, we introduce a contrastive loss that circumvents the inherently discontinuous and non-differentiable nature of QUBO solvers. Our method is compatible with any QUBO solver at training and test time—we consider parallelised exhaustive search, simulated annealing, and quantum annealing. QUBO learning, *i.e.,* determining a function returning QUBO forms given a problem instance of some problem type as input, is a non-trivial and challenging task. In summary, this paper makes several technical contributions to enable QUBO learning:

1. QuAnt, *i.e.,* a new meta-learning approach to obtain QUBO forms executable on modern QAs for computer vision problems. While prior methods rely on analytical derivations, we learn QUBOs from data (Sec. 6.4.1).

2. A new training strategy for neural methods with backpropagation involving finding low-energy solutions to instantaneous (optimised) QUBO forms, independent of the solver

(Secs. 6.4.2 and 6.4.3).

3. Application of the new framework to several problems with solutions encoded by permutations and discretised rigid transformations (Secs. 6.4.4 and 6.4.5).

We show that our methodology is a standardised way of obtaining QUBOs independent of the target problem type. This paper focuses on three problem types already tackled by QCV methods relying on analytical QUBO derivations: graph matching and point set alignment (with and without known prior point matches in the 3D and 2D cases, respectively). We emphasise that we do not claim to outperform existing specialised methods for these problem types or that QA is particularly well-suited for them. Rather, we show that this wide variety of problems can be tackled successfully and competitively by our general quantum approach already now, before quantum primacy. Thus, in the future, computer vision methods may readily benefit from the (widely expected) speed-up of QC through an *easy* and *flexible* re-formulation of algorithmic parts as QUBOs, thanks to our proposed method. We run our experiments on D-Wave Advantage5.1 [84], an experimental realisation of AQC with remote access. This paper assumes familiarity with the basics of quantum computing. For convenience, we summarise several relevant definitions in the Appendix.

6.3 Related Work

The two main paradigms for quantum computing are *gate-based* QC and adiabatic quantum computing (AQC). Our method uses quantum annealing, which is derived from AQC, and is *not* gate-based. The predominantly theoretical field of quantum machine learning (QML) investigates how quantum computations can be integrated into machine learning [38, 92, 242, 124, 91, 174, 155]. Many QML methods assume gate-based quantum computers and define a quantum variational layer, *i.e.,* a sequence of parametrised unitary transformations meant for execution on quantum hardware [188]. QML methods are often optimised using variants of the backpropagation algorithm [180, 264, 32]. Quantum variational models were recently applied to combinatorial optimisation [142] and reinforcement learning [93, 168]. Instead of learning to regress unitary transformation parameters for gate-based QC, we learn to regress QUBO forms for QA.

In contrast to gate-based machines, QAs can already solve real problems formulated as QUBOs [191, 252, 247, 204, 179, 246]. Recently, QCV has rapidly transitioned from theoretical considerations [263, 61, 194, 192] to practical algorithms leveraging quantum-mechanical effects of quantum computers, ranging from image retrieval and processing [263, 272], classification [45, 196, 58, 268, 85] and tracking [163, 275], to problems on graphs [279, 237, 174], consensus maximisation [90], shape alignment [198, 238], segmentation [18] and ensuring cycle-consistency [40, 274].

Many of these methods are evaluated on real quantum hardware, as both gate-based and QA machines can be accessed remotely [71, 221].

We demonstrate the efficacy of our QuAnt approach on the applications of graph matching and point set alignment where we compare against recent quantum state-of-the-art methods [237, 111], respectively.

Another line of work in different domains concerns learning the best adiabatic quantum algorithm. While some works [212, 211] develop an algorithm inspired by tabu search, our

method uses a neural network to output a coupling matrix. Orthogonal to our work, others train neural networks to solve problem-specific QUBOs [106]. [200] optimize a blackbox function by finding a QUBO as surrogate model. QML on gate-based QC has been studied at length, but machine learning with QA remains largely under-explored, with only a few exceptions (*e.g.,* linear regression [82] and binary neural networks [233]). In stark contrast to existing QCV methods with analytically derived QUBOs [163, 237, 238, 40], our approach enables more flexible and compact solution encodings.

QuAnt is also related to recent non-quantum approaches that aim to improve combinatorial optimisation by seamlessly integrating deep learning and combinatorial building blocks as custom layers and backpropagating through them [102, 224, 266]. In this respect, ours is the first work that uses a quantum QUBO solver in neural architectures.

6.4 QUBO Learning Approach

We present a new meta-learning approach for regressing quadratic unconstrained binary optimisation problems (QUBOs) suitable for modern quantum annealers (QA); see Fig. 6.1. While existing works analytically derive QUBOs for different problems [40, 237, 238, 275], we propose to instead *learn* a function that turns a problem instance into a QUBO to be solved by a QA. Specifically, we train a multi-layer perceptron (MLP) that takes a vectorised problem instance and regresses the QUBO weights such that the QUBO minimiser is the solution to the problem. Note that we only specify the bit encoding of the solution but let the network learn to derive QUBOs. Crucially, we show how training the MLP is possible despite quantum annealing (like any QUBO solver) being discontinuous and non-differentiable.

6.4.1 QUBOs and Quantum Annealing

Quantum annealing is a metaheuristic to solve $\mathcal{NP}$-hard problems of the form $\arg\min_{\mathbf{s}\in\{-1,1\}^n} \mathbf{s}^\top \mathbf{J}\mathbf{s} + \mathbf{b}^\top\mathbf{s}$, where $\mathbf{s}$ is a binary vector, $\mathbf{J}\in\mathbb{R}^{n\times n}$ is a matrix of *couplings*, and $\mathbf{b}\in\mathbb{R}^n$ contains *biases* [182]. We can rewrite this as a QUBO: $\arg\min_{\mathbf{x}\in\{0,1\}^n} \mathbf{x}^\top\mathbf{A}\mathbf{x}$, by substituting $\mathbf{x} = \frac{1}{2}(\mathbf{s}+\mathbb{1}_n)$ and $\mathbf{A} = \frac{1}{4}\mathbf{J} + \frac{1}{2}(\mathbf{J}\mathbb{1}_n + \mathbb{1}_n^\top\mathbf{J}) + \frac{1}{2}\mathrm{diag}(\mathbf{b})$.

In quantum annealing, the binary n-dimensional vector $\mathbf{x}$ describes the measurement outcomes of n qubits. Annealing starts out with an equal superposition quantum state of the qubits that assigns an equal probability to all possible binary states $\{0,1\}^n$. During the anneal, the couplings and biases of $\mathbf{A}$ are gradually imposed on the qubits. The adiabatic theorem [44] implies that doing so sufficiently slow forces the qubits into a quantum state that assigns nonvanishing probability only to binary states that minimise the QUBO [101]. We then only need to measure the qubits to determine their binary state, which is our solution $\mathbf{x}$. For a more detailed description of quantum annealing, we refer to prior computer-vision works [237, 238, 111, 163] and Appendix 6.7.

6.4.2 Network Architecture and Losses

In this section, we describe the network architecture that takes as input a problem instance and regresses a QUBO whose solution (*e.g.,* obtained via quantum annealing) solves the problem instance. For a given problem type, we require a problem description that is amenable to

QUBO learning: A parametrisation of problem instances as real-valued vectors $\mathbf{p} \in \mathbb{R}^m$, and a parametrisation of solutions as binary vectors $\mathbf{x} \in \{0,1\}^n$. Since we use supervised training, we additionally need a training set $\mathcal{D} = \{(\mathbf{p}_d, \hat{\mathbf{x}}_d)\}_{d=1}^D$ containing D problem instances $\mathbf{p}_d$ with ground-truth solutions $\hat{\mathbf{x}}_d$.

We use a multilayer perceptron (MLP) with L layers and H hidden dimensions, ReLU activations (except for the last layer, which uses sin [244]), and concatenating skip connections from the input into odd-numbered layers (except for the first and last layers). The input to the network is a problem instance $\mathbf{p}$, and the output is a QUBO matrix $\mathbf{A}$: $\mathbf{A} = \mathrm{MLP}(\mathbf{p})$.

We could now use a dataset of problem instances and corresponding $\mathbf{A}$ to supervise the MLP directly. However, this requires specifying how $\mathbf{A}$ is to be derived for a certain instance, which comes with two downsides: (1) A problem-type-specific algorithm for analytically deriving instance-specific $\mathbf{A}$ needs to be designed to generate enough training data $\{(\mathbf{p}_d, \mathbf{A}_d)\}_{d=1}^D$, which is non-trivial, and (2) The binary parametrisation ($\mathbf{x}$) of the solution space depends on the algorithm, which can lead to more variables than intrinsically needed by the problem (e.g., if $\mathbf{x}$ needs to represent one of k numbers, a one-hot parametrisation would have length $n{=}k$, while a binary-encoding parametrisation would have length $n{=}\log k$). This is particularly problematic as contemporary quantum hardware only provides a limited number of qubits. We thus choose to supervise $\mathbf{A}$ not directly and, instead, supervise the solutions of the QUBO. This strategy tackles both issues as it lets the network *learn* an algorithm *compatible* with the (potentially shorter) solution parametrisation. Therefore, our method is easily applicable to new problem types, as we show in Secs. 6.4.4 and 6.4.5.

The regressed $\mathbf{A}$ defines a QUBO, which can be solved by any QUBO solver. But how can we, during training, backpropagate gradients from the solution binary vector $\mathbf{x}$ through the QUBO solver despite these solvers having zero gradients almost everywhere [266]? We circumvent this issue by exploiting a contrastive loss (*cf.* [160, Eq. (10)]) as follows: We know the energy of the ground-truth solution $\hat{\mathbf{x}}$ of the problem instance, namely $\hat{\mathbf{x}}^\top \mathbf{A} \hat{\mathbf{x}}$. If the energy of the minimiser $\mathbf{x}^* = \mathbf{x}^*(\mathbf{A})$ of the current QUBO is lower, then $\mathbf{A}$ does not yet describe a QUBO that outputs the right solution. We, therefore, seek to push the energy of $\hat{\mathbf{x}}$ lower while pulling the energy of the minimiser $\mathbf{x}^*$ up:

$$L_{\text{gap}} = \hat{\mathbf{x}}^\top \mathbf{A} \hat{\mathbf{x}} - \mathbf{x}^{*\top} \mathbf{A} \mathbf{x}^*, \tag{6.1}$$

which has a zero gradient if $\mathbf{x}^* = \hat{\mathbf{x}}$, as desired. L_{gap} avoids backpropagation through $\mathbf{x}^*$ and is even compatible with automatic differentiation. It yields the following update for an entry of $\mathbf{A}$, which can then be further backpropagated into the MLP via the chain rule:

$$\frac{\partial L_{\text{gap}}(\mathbf{A})}{\partial \mathbf{A}_{i,j}} = 2\hat{\mathbf{x}}_i \hat{\mathbf{x}}_j - 2\mathbf{x}^*_i \mathbf{x}^*_j - 2\frac{\partial \mathbf{x}^*(\mathbf{A})}{\partial \mathbf{A}_{i,j}} \mathbf{A} \mathbf{x}^*, \tag{6.2}$$

where the last term comes from the dependency of $\mathbf{x}^*$ on $\mathbf{A}$ via the QUBO solver. While intuitively useful, we note that this term is zero "almost everywhere" (in the mathematical sense), and we hence ignore it as it provides almost no information. Note that this is a common approximation (*e.g.,* auto-differentiation frameworks backpropagate through $\max(\cdot)$ pooling in the same manner).

Unfortunately, L_{gap} alone would not prevent degenerate $\mathbf{A}$, which have multiple solutions, including undesirable ones. We, therefore, discourage such $\mathbf{A}$ that have more than one solution

$\mathbf{x}^*$:

$$L_{\text{unique}} = -|\hat{\mathbf{x}}^\top \mathbf{A}\hat{\mathbf{x}} - \mathbf{x}^{+\top}\mathbf{A}\mathbf{x}^+|, \tag{6.3}$$

where $\mathbf{x}^+$ is the $\mathbf{x}$ that minimises $\mathbf{x}^\top\mathbf{A}\mathbf{x}$ over $\{0,1\}^n \setminus \{\mathbf{x}^*\}$, and $|\cdot|$ is the absolute value operator.

In addition to these data terms, we found it helpful to regularise the network by encouraging sparsity on all intermediate features of the MLP [165]:

$$L_{\text{mlp}} = \sum_{\mathbf{f}\in\mathcal{F}} \frac{1}{|\mathbf{f}|}\|\mathbf{f}\|_1, \tag{6.4}$$

where $\mathcal{F}$ is the set of all network layer outputs (except for the last layer). The total loss then reads:

$$L = L_{\text{gap}} + \lambda_{\text{unique}} L_{\text{unique}} + \lambda_{\text{mlp}} L_{\text{mlp}}, \tag{6.5}$$

where we set $\lambda_{\text{mlp}} = 10^{-4}$ and $\lambda_{\text{unique}} = 10^{-3}$ regardless of problem type.

6.4.3 QUBOs on D-Wave Quantum Annealers

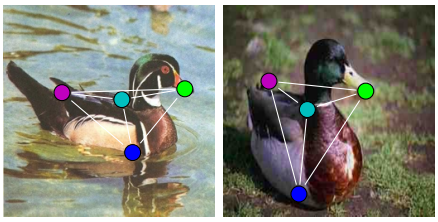
We can use D-Wave to solve any generated QUBO $\mathbf{A}$, where $\mathbf{A}_{i,j} \in \mathbb{R}$ describes the direction and coupling strength between logical qubits i and j. However, each physical qubit in the annealer is only connected to a small subset of other physical qubits, which makes the regressed $\mathbf{A}$ not directly compatible with the annealer. We tackle this issue by manually pre-determining a sparse connectivity pattern of the physical qubits and then masking out the other entries of $\mathbf{A}$ before solving, such that the MLP focuses on only these sparse entries. For example, when $\mathbf{x} \in \mathbb{R}^n$ with n=8, we can use D-Wave's Chimera architecture, which is made up of interconnected $\mathcal{K}_{4,4}$ unit cells [84]. ($\mathcal{K}_{4,4}$ is a complete bipartite graph with two sets of four qubits each.) Since n=8, we can fit one problem instance into one unit cell, which allows us to anneal many problem instances in parallel by putting them in different unit cells. This speeds up training and saves expensive time on the quantum annealer. For larger problems with n=32, we can use D-Wave's Pegasus architecture [42], which has more interconnections (qubit couplings) between its $\mathcal{K}_{4,4}$ unit cells than Chimera. We use four such unit cells per problem instance, following D-Wave's pattern [7]. See Fig. 6.4b for an exemplary colour-coded qubit connectivity pattern of $\mathbf{A}$.

Given our full method, we next show how it can be applied to three problem types, *i.e.,* graph matching, point set registration, and rotation estimation; see Appendix for further details.

6.4.4 Graph Matching

The goal of graph matching is to determine correspondences from k key points in two images or graphs; see Fig. 6.2 for an example. This can be formalised as a quadratic assignment problem with a permutation matrix representing the matching [237]: $\arg\min_{\mathbf{X}\in\mathbb{P}_k} \mathbf{x}^\top\mathbf{W}\mathbf{x}$, where $\mathbb{P}_k$ is the set of permutation matrices, $\mathbf{x} = \text{vec}(\mathbf{X}) \in \{0,1\}^{k^2}$ is the vectorised permutation matrix, and $\mathbf{W} \in \mathbb{R}^{k^2\times k^2}$ contains pairwise weights. Unfortunately, the permutation constraint cannot be directly realised on the quantum annealer.

Instead, note that a permutation $P : [k] \to [k]$ is fully defined by the sequence $P(1), P(2), \ldots, P(k)$. Our method can use an efficient binary encoding for each entry of this sequence, using only $k \log k$ binary variables in total. Note that not all vectors $\mathbf{x} \in \{0,1\}^{k \log k}$ are valid permutations. As an optional post-processing step, we can perform a *projection* to the nearest permutation with respect to the Hamming distance in our binary encoding. Unless stated otherwise, we do not apply this post-processing.



(a) Source (b) Target

Figure 6.2: Example matching on four key points from the Willow dataset [62]. Corresponding points (same colours) are found based on feature similarity.

In addition to the solution parametrisation, we also need to design the problem description $\mathbf{p}$. For real data, we use $\mathbf{p} = \text{vec}(\mathbf{W})$, where the diagonal contains cosine similarities between the feature vectors extracted with AlexNet [154] pre-trained on ImageNet [88] of all key point pairs, and the off-diagonal follows the geometric term from [256, Eq. (6)]. For evaluation, we also introduce the synthetic *RandGraph* dataset; it uses matrices of random distances $\mathbf{D} \in \mathbb{R}^{k \times k}$ with entries $\mathbf{D}_{i,j} \in \mathcal{U}(0,1)$ to define $\mathbf{W}_{k \cdot i + P(i), k \cdot j + P(j)} = \left| \mathbf{D}_{i,j} - \mathbf{D}_{P(i),P(j)} \right|$. The MLP thus learns to compress the input matrix into a much smaller QUBO.

6.4.5 Point Set Registration and Rotation Estimation

In 2D point set registration, we are given two point sets with potentially different numbers of points and no correspondences, and we seek to find a rotation angle that best aligns them. We follow Golyanik *et al.* [111] and use the vectorised form of their input matrix to represent a problem instance. We parametrise the solution space of $\mathbf{x} \in \{0,1\}^9$ by splitting the output space $[0, \frac{1}{3}\pi]$ into 2^9 equally sized bins and consecutively indexing them with a 9-bit integer.

In 3D rotation estimation, we are given two 3D point clouds with known matches and seek to estimate the 3D rotation aligning them. We represent a problem instance by the vectorised covariance matrix of the two point clouds; 3D rotation is parametrised by Euler angles α, β, γ. We discretise each angle into 2^5 bins, such that $\mathbf{x} \in \{0,1\}^{15}$.

6.5 Experimental Evaluation

We next experimentally evaluate QuAnt. Our goal is to show that it outperforms the previous quantum state of the art. For reference, we also report comparisons against specialised classical methods.

Data. We evaluate graph matching on the Willow object dataset [62], which contains labelled key points. We use k=4 randomly chosen key point pairs per image. We use 5640 images for training, and test on 846 images. Both of the sets are obtained via `pygmtools` [254]. We also evaluate on our synthetic dataset *RandGraph* (see Sec. 6.4.4), with both k=4 and k=5. We evaluate 2D point set registration on the 2D Shape Structure dataset [56] providing 2D silhouette images of real-world objects. We treat the silhouette outlines as 2D points. We use 500 shapes from various classes for training, and test on 50 shapes. For each shape, we apply 1000 (for train) or 100 (for test) different rotations of up to 60° and pick random pairs to generate problem instances. For 3D rotation estimation, we evaluate on ModelNet10 [270], which contains CAD models of ten

object categories. We proceed with point cloud representations of each shape. We use 300 shapes from various classes for training, and test on 30 shapes from various classes. For each shape, we apply 1000 different 3D rotations with angle ranges $\alpha, \gamma \in [-\frac{1}{9}\pi, \frac{1}{9}\pi]$ and $\beta \in [-\frac{1}{18}\pi, \frac{1}{18}\pi]$ and pick random pairs to generate problem instances.

Comparisons. We compare QuAnt to two baselines and specialised methods, depending on the problem type. For all problem types, we demonstrate the power of using QUBOs compared to the *Diag* baseline that regresses a diagonal QUBO matrix $\mathbf{A}$ (which is trivially solvable). While this baseline ablates the QUBO itself, we also consider a more natural neural network baseline, *i.e., Pure*, that regresses the binary solution directly (there is no activation after the last layer) and uses an ℓ_1-loss between the output and $\hat{\mathbf{x}}$ instead of L_{gap} and L_{unique}. At test time, we threshold the network output of Pure at 0 to obtain binary vectors. For QuAnt, Diag and Pure variants, we experiment with all combinations of the numbers of layers $L \in \{3, 5\}$ and hidden dimensions $H \in \{32, 78\}$.

We compare our graph matching results with the *Direct* baseline on Willow [62]; we directly solve the quadratic assignment problem given by $\mathbf{W}$ with exhaustive search, which provides an upper bound for our method. We also compare against Quantum Graph Matching (QGM) [237], to which we pass our input matrices $\mathbf{W}$. For 2D point set registration, we compare against the analytic quantum method (AQM) [111], which is an upper bound for our technique since we take its vectorised QUBO as input, and against the classical, specialised ICP algorithm [170]. For 3D rotation estimation, we use Procrustes as a classical specialised method, which is thus an upper bound for our (general) method.

Metrics. We measure accuracy of the graph matching solutions as the percentage of correctly recovered permutation matrices. For 2D point set registration and 3D rotation estimation, we quantify the difference between the known ground-truth rotations and the estimated rotations by their geodesic distances (angles) in the rotation groups $SO(2)$ and $SO(3)$, respectively.

QUBO Solvers. For graph matching, we follow Sec. 6.4.3 to make our regressed QUBOs compatible with the QA. Due to a restricted QA compute budget, we train and test with simulated annealing unless stated otherwise. For the point cloud experiments, we regress dense $\mathbf{A}$ and use our exhaustive search implementation at train and test time unless stated otherwise. When evaluating on the QA, we rely on minor embeddings to make the regressed $\mathbf{A}$ compatible with the QA. Please refer to the Appendix for the details.

6.5.1 Results

General Baselines. The quantitative results for graph matching, point set registration, and rotation estimation are reported in Tables 6.1, 6.2, and 6.3, respectively. Across network sizes and all three problem types, the results show that having a full, $\mathcal{NP}$-hard QUBO (ours) instead of only a diagonal QUBO (Diag) is advantageous. We also find that the proposed method yields better results than Pure on both point set registration and rotation estimation, although Pure yields better results for graph matching.

Specialised Methods. For reference, we compare QuAnt to methods specialised to a certain problem type. Since our approach is general, they mostly provide an upper bound for our performance.

We evaluate QGM [237] on several RandGraph instances. We confirm their finding that strongly enforcing permutation constraints eventually retrieves the right permutation as the

Table 6.1: Comparison to general baselines on graph matching. We report the accuracy (in %).

(a) RandGraph for k=4.

	Ours	Diag	Pure
L=3, H=32	9	8	**91**
L=3, H=78	30	18	**96**
L=5, H=32	11	11	**89**
L=5, H=78	49	43	**96**

(b) Willow object dataset [62] for k=4. Trained for 300 epochs with L=5, H=78.

Ours	Diag	Pure	Direct
69	53	90	**97**

Table 6.2: Comparison to general baselines on point set registration. We report averages of the mean errors and their standard deviations over three runs.

	Ours	Diag	Pure
L=3, H=32	8.4 ± 0.8	11.1 ± 1.3	**8.2 ± 1.2**
L=3, H=78	**7.2 ± 1.1**	8.3 ± 0.7	9.3 ± 1.9
L=5, H=32	**8.6 ± 0.5**	10.9 ± 1.2	9.3 ± 1.9
L=5, H=78	**6.8 ± 0.3**	7.7 ± 0.5	11.3 ± 4.5

Table 6.3: Comparison to general baselines on rotation estimation. We report averages of the mean errors and their standard deviations over three runs.

	Ours	Diag	Pure
L=3, H=32	5.9 ± 3.0	**5.4 ± 1.0**	7.9 ± 0.5
L=3, H=78	**4.1 ± 0.5**	5.0 ± 0.3	7.1 ± 0.1
L=5, H=32	**3.7 ± 0.8**	5.0 ± 0.4	16.2 ± 7.1
L=5, H=78	**3.4 ± 0.4**	4.7 ± 0.2	10.1 ± 1.8

sample with the lowest energy. However, using the analytical bound for the penalty term leads to a success probability (*i.e.,* the probability of getting the best solution across anneals) smaller than random guessing due to experimental errors in the couplings. Next, we find that the QUBOs of QuAnt are much smaller and better suited to be solved with a quantum annealer than QGM's. For RandGraph with k=5, our method needs 15 physical qubits while their baseline and row-wise methods need 89 qubits on average and a chain length of four, and their *Inserted* method needs, on average, 39 qubits and a chain length of three on D-Wave Advantage. Thus, our success probability of 26% when evaluating on test data is orders of magnitude higher than *Inserted*'s 0.22% (best in QGM). This shows how QuAnt improves over the quantum state of the art even though we merely focus on the solution with the lowest energy across anneals, while they focus on the success probabilities. We refer to Appendix 6.10 for a detailed evaluation. Table 6.1b confirms that *Direct* is an upper bound to our approach.

Table 6.8 shows quantitative results for 2D point set matching without noise. AQM slightly outperforms QuAnt, which is expected as we take AQM's QUBO as input; hence, its performance is an upper bound for our method. Fig. 6.3 shows a qualitative example.

As expected, quantitative results in Table 6.7 (with no incorrect correspondences) show that classical, specialised Procrustes performs better than our general method on 3D rotation estimation. Note that our technique yields better results than Procrustes under test-time noise, as we discuss later in detail.

6.5.2 Ablation

We ablate L_{unique} and L_{mlp} in Table 6.4. For graph matching, we use RandGraph with k=4. We find that removing either L_{unique} or L_{mlp} leads to mixed results on graph matching and worse results in almost all cases for points set registration and rotation estimation.

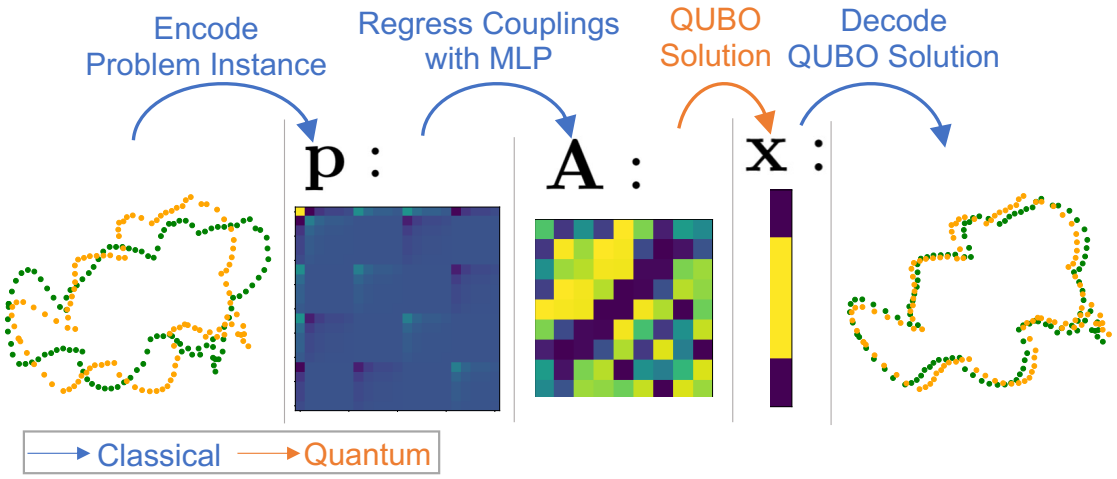


Figure 6.3: Test-time example inputs and outputs of QuAnt trained for 2D point set registration.

Table 6.4: Loss ablations. We report accuracy for graph matching (in %) and mean/median error otherwise.

	Graph Matching			Point Set Registration			Rotation Estimation		
	w/o L_{unique}	w/o L_{MLP}	Ours	w/o L_{unique}	w/o L_{MLP}	Ours	w/o L_{unique}	w/o L_{MLP}	Ours
L=3, H=32	7	**9**	**9**	17.8 / 12.0	18.6 / 13.4	**15.0** / **8.7**	5.1 / 5.0	3.4 / 3.0	**3.4** / **3.0**
L=3, H=78	**30**	29	**30**	15.5 / 8.0	21.8 / 17.0	**14.5** / **7.7**	**2.9** / **3.0**	4.6 / 5.0	4.2 / 4.0
L=5, H=32	**14**	6	11	18.1 / 11.7	19.0 / 7.7	**9.0** / **4.6**	3.4 / 3.0	2.5 / 2.0	**2.3** / **2.0**
L=5, H=78	46	**54**	49	18.3 / 11.7	**17.8** / 11.7	18.5 / **7.7**	3.7 / 4.0	4.1 / 4.0	**3.3** / **3.0**

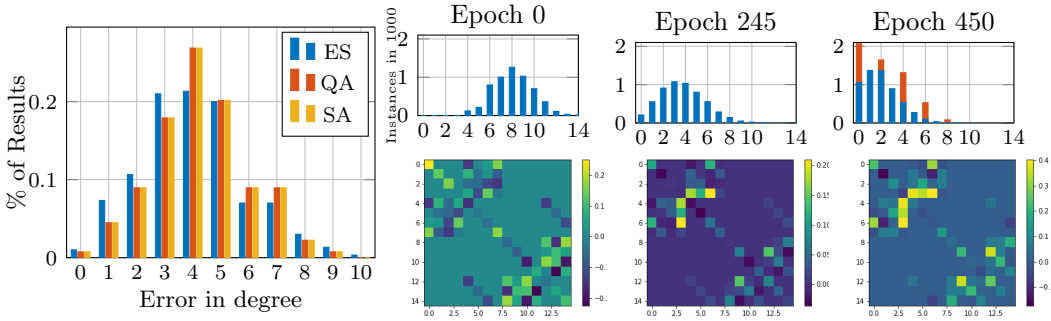
6.5.3 Evaluation on D-Wave

By design, our QuAnt is agnostic to the type of QUBO solver used. After training with exhaustive search, we compare how the performance on the test set differs under exhaustive search, SA, or QA. The results in Fig. 6.4a show that the exact solutions of exhaustive search only slightly outperform the less computationally expensive QA and SA. Moreover SA yields results very similar to QA.

Next, we compare the test-time results of QA and SA (after training the method with the same technique, QA and SA, respectively). See Table 6.5 for the results, both with and without projecting the final binary solution to a valid permutation encoding during post-processing. Training with QA delivers better results than training with SA. We attribute this to a better second-best solution $\mathbf{x}^+$ used by L_{unique}. While SA yields solutions $\mathbf{x}^*$ that are comparable to QA, its second-best solutions are worse than QA's. We refer to the appendix for details. Unfortunately, real-world compute resources for training with QA remain limited, as of this writing. We, therefore, fall back on SA for larger-scale experiments in this work. However,

Table 6.5: QuAnt (L=5, H=78) on RandGraph (k=5) trained for 450 epochs (QA or SA).

	SA	QA
Before projection	10	**18**
After projection	24	**36**



(a) Errors on rotation estimation. (b) Evolution of result (top) and coupling matrix (bottom).

Figure 6.4: (a) Histogram of errors for rotation estimation using exhaustive search (ES), quantum annealing (QA), and simulated annealing (SA) at test time. The maximum error on the x-axis amounts to 59 and no methods have higher errors than shown. (b) Evolution over different epochs of the Hamming distance between predicted solutions and the ground truth (top), and coupling matrix when training our approach for graph matching (bottom). (Top): The x-axis shows the Hamming distance. Blue indicates unprojected results, and red means after projection to a permutation. We only project after training.

Table 6.5 suggests that our results could improve noticeably on QA.

6.5.4 Further Analysis

Training. Fig. 6.4b visualises how the instantaneous solution and **A** matrix evolve for graph matching.

Varying Problem Difficulty. We provide a more detailed analysis of the performance of our method on point set registration for varying difficulty levels. Table 6.6 shows that a larger input misalignment between the two point clouds worsens the results, as expected.

Table 6.6: Interval analysis for point set registration with L=5, H=78. We evaluate on point cloud pairs with ground-truth angles uniformly sampled within the given intervals. We report the mean/median error in degrees.

angle interval	$0 - 1\frac{\pi}{18}$	$1\frac{\pi}{18} - 2\frac{\pi}{18}$	$2\frac{\pi}{18} - 3\frac{\pi}{18}$	$3\frac{\pi}{18} - 4\frac{\pi}{18}$	$4\frac{\pi}{18} - 5\frac{\pi}{18}$	$5\frac{\pi}{18} - 6\frac{\pi}{18}$
mean / median	4.0 / 1.9	5.1 / 2.6	6.4 / 3.0	7.9 / 3.8	7.9 / 3.8	8.4 / 4.0

Robustness to Noise. We investigate the robustness of our method and other approaches against input noise at test time after training without noisy data. We look at rotation estimation, where we randomly pick a fixed percentage of points and randomly permute their correspondences (among themselves). Table 6.7 contains results. The quality of QuAnt's results barely degrades with increasing noise levels, even for 20% of incorrect correspondences. QuAnt already outperforms the classical Procrustes for even 1% of incorrect correspondences, even though Procrustes also starts from the same covariance matrix. We observe that the advantage of our method grows with larger noise levels. QuAnt also consistently performs better than the general baselines, which are similarly robust to increasing noise levels.

We next look at point set registration under input noise at test time and after training

Table 6.7: Robustness to varying amounts of incorrect test-time correspondences in rotation estimation. We report mean/median error for L=3, H=32. The first column specifies the percentage of incorrect correspondences at test time.

%	**Ours**	Procrustes	Diag	Pure
0	3.9 / 4.0	**0.0** / **0.0**	5.6 / 6.0	8.1 / 8.0
1	**3.4** / **3.0**	5.8 / 3.0	5.7 / 6.0	8.2 / 8.0
5	**3.4** / **3.0**	25.7 / 13.0	6.0 / 6.0	8.2 / 8.0
10	**3.2** / **3.0**	43.8 / 21.0	6.2 / 6.0	8.2 / 8.0
15	**3.5** / **3.0**	64.7 / 58.0	6.2 / 6.0	8.2 / 8.0
20	**3.7** / **3.0**	75.3 / 79.0	5.8 / 6.0	8.2 / 8.0

Table 6.8: Robustness to varying amounts of uniform noise in point set registration. We report mean/median error for L=5, H=78 and the number of logical/physical qubits. The first column states the range of the noise in % of the maximum extent of the point cloud. "†": uncoupled qubits.

%	**Ours**	Diag	Pure	AQM
0	5.8 / 3.5	7.3 / 4.7	6.8 / 5.9	**4.3** / **2.6**
5	6.4 / 3.3	7.0 / 5.2	7.0 / 6.1	**4.5** / **2.9**
10	6.5 / **3.3**	8.4 / 5.2	7.1 / 6.5	**5.6** / 3.8
15	7.2 / **3.5**	9.5 / 5.9	7.9 / 6.7	**5.6** / 3.8
20	10.3 / 5.4	11.6 / 6.6	8.2 / 6.8	**5.9** / **3.3**
Qubits	**9/14**	(9/9)†	n/a	21/∼55

without noise. Here, we add uniform noise to one point cloud, where the range of the noise is a percentage of the maximum extent of the point cloud. Table 6.8 contains the results. ICP, an iterative approach, is robust to the noise, gives highly accurate results and, thus, outperforms the competing non-iterative approaches. Since QuAnt takes as input the vectorised QUBO that AQM solves, AQM constitutes an upper bound for the performance of our approach. However, QuAnt could, in principle, scale to 3D point set matching while AQM's solution parametrisation severely inhibits scaling to larger problems. Finally, QuAnt performs better than the general baselines.

6.6 Discussion

Limitations and Future Work. As all learning-based approaches, QuAnt can perform worse on problem instances that fall significantly outside the training distribution. While our general method does not outperform classical methods specialised on certain problem types, we achieve performance on par with hand-crafted QUBO designs used in state-of-the-art QCV methods. We achieve this while greatly reducing the effort required for new problem types. For our point cloud experiments, we rely on minor embeddings to transfer the regressed dense QUBOs to the QA. On existing hardware, large minor embeddings can worsen the resulting quality noticeably. However, we only need to embed a QUBO with nine logical qubits into 14 physical qubits. Although our focus is on a general design, our core idea of learning QUBOs can be specialised to any given problem type by designing a more specific network architecture and losses that capture priors for the problem type.

Conclusion. We showed that learning to regress QUBO forms for different problems instead of deriving them analytically can be a reasonable alternative to existing methods. We showed

the generality of QuAnt on diverse problem types. Our experiments demonstrated that learning QUBO forms and solving them either on a quantum annealer or with simulated annealing, in most cases, leads to better results than directly regressing solutions. Moreover, QuAnt significantly outperformed the previous quantum state of the art in graph matching and rotation estimation in the setting with noise. We believe our work considerably broadens the available toolbox for development and analysis of quantum computer vision methods and opens up numerous avenues for future research.

Appendix

This appendix provides more details on adiabatic quantum computing in Sec. 6.7. We provide training and implementation settings in Sec. 6.8. Further details on the problem description for graph matching and a failure case are in Sec. 6.9. Sec. 6.10 contains a deeper comparison with QGM [237] and Sec. 6.11 compares SA and QA. In Sec. 6.12, we provide further quantitative and qualitative results on rotation estimation. Finally, Sec. 6.13 contains more details and experiments on point set registration.

6.7 Quantum Computing Background

6.7.1 Quantum Annealing in Detail

As we have seen, quantum annealing is a metaheuristic to solve the $\mathcal{NP}$-hard Ising problem:

$$\underset{\mathbf{s}\in\{-1,1\}^n}{\arg\min}\ \mathbf{s}^\top \mathbf{J}\mathbf{s} + \mathbf{b}^\top \mathbf{s}, \tag{6.6}$$

where $\mathbf{s}$ is a binary vector, $\mathbf{J} \in \mathbb{R}^{n\times n}$ is a matrix of *couplings*, and $\mathbf{b} \in \mathbb{R}^n$ contains *biases* [182]. Here, we give a brief overview of how this fits in the framework of quantum mechanics. D-Wave quantum annealers rely on magnetic fluxes in superconducting Niobium loops [203]. The direction of the current flowing through them can be modelled as a qubit, *i.e.,* as a two-dimensional, complex, normalised vector $|\psi\rangle \in \mathbb{C}^2$ in the Dirac notation. In the so-called computational basis, the basis vectors correspond to the current flowing clockwise or anti-clockwise. After measuring the state, the system will collapse to either basis state. The absolute value of the complex-valued coefficients of the linear combination (probability amplitudes) is the probability of each outcome after measurement (*e.g.,* clockwise or anti-clockwise current). The state space of $n \in \mathbb{N}$ qubits can be expressed with the tensor product $\bigotimes_{i=1}^{n} \mathbb{C}^2$ and is thus a 2^n-dimensional complex vector space. We need that many parameters because *entangled* states cannot be described separately. If the two states $|\psi\rangle, |\phi\rangle$ corresponding to different physical systems (*e.g.,* two niobium loops or two atoms) can be described independent from each other, the whole system is described by $|\psi\rangle \otimes |\phi\rangle$. Note that if every state could be decomposed this way, one would only need $2n$ parameters.

The evolution of a quantum state $|\psi\rangle$ over time can be described with the time-dependent Schrödinger equation:

$$H|\psi\rangle = i\hbar \frac{\partial}{\partial t}|\psi\rangle, \tag{6.7}$$

where the Hamilton operator H is a Hermitian Matrix describing the possible energies of the system, i is the imaginary unit, $\hbar$ is a constant, and t denotes time. For adiabatic quantum computing, one needs a problem Hamiltonian H_P, where the eigenvector corresponding to the lowest eigenvalue is a solution to the particular Ising problem, and an initial Hamiltonian H_I with an easy-to-prepare ground state. The Adiabatic Theorem [44] states that if we start with the ground state of H_I and take a sufficiently long time τ to gradually change from H_I to H_P, *e.g.,* with:

$$H(t) = (1 - \frac{t}{\tau})H_I + \frac{t}{\tau}H_P, \tag{6.8}$$

then we end up in the ground state of H_P. From the latter, we can deduce the solution of the particular Ising problem. Simulating this whole process classically can be difficult (or even intractable) because we are dealing with $2^n \times 2^n$ matrices H_I and H_P, where n is the number of qubits. How difficult the classical simulation is, depends on the exact form of the Hamiltonians. (Particularly promising for speed-ups are, *e.g.,* so-called non-stoquastic Hamiltonians [11].)

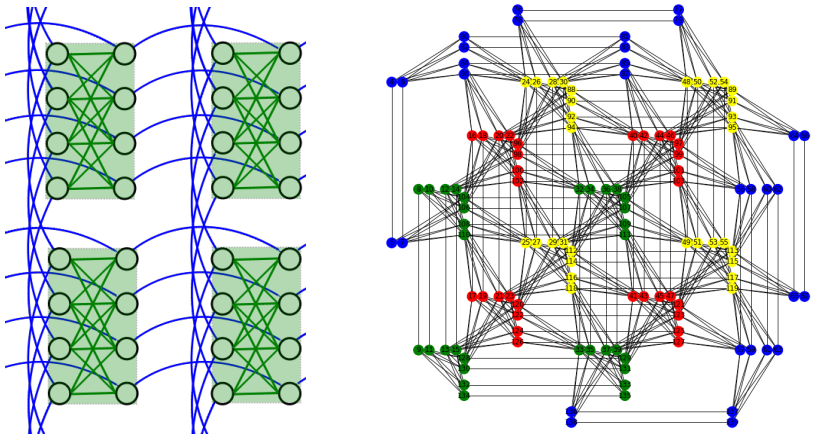
6.7.2 Logical and Physical Qubits

The QUBO defines the couplings between two *logical* qubits i and j. Such a QUBO can contain couplings between any two qubits. However, in contemporary hardware realisations, each *physical* qubit is only connected to a few others (see Fig. 6.5a and Fig. 6.5b). In the main paper, we show how the QUBO matrix $\mathbf{A}$ can account for this sparse connectivity pattern by setting entries between logical qubits i and j to 0 if the physical qubits i and j have no connection. Still, D-Wave supports denser connectivity patterns than what is implied by the hardware: Multiple physical qubits can be chained together to represent a single logical qubit of $\mathbf{x}$ that has many connections. The physical qubits in the chain will then have strong couplings along the chain to encourage them to all end up in the same final state (either all 0 or all 1), representing the final state of the corresponding logical qubit. This is formalised as a minor embedding (of the connectivity graph of the logical qubits) into the connectivity graph of the physical qubits. Using the heuristic method of Cai and colleagues [52] is popular to determine the minor embeddings in practice.

6.8 Implementation Details

Our code, which we will release, is implemented in Pytorch [213]. We use Adam [148] with a learning rate of 10^{-3} for training. For graph matching on RandGraph with k=4, we use a batch size of 141 and train for 150 epochs, which takes about seven hours. For RandGraph k=5, we use 450 epochs, which takes about 23 hours. For Willow, we train for 300 epochs, which takes about 14 hours. The baselines are trained for the same number of epochs. While Diag takes a comparable amount of time, the Pure baseline takes about three minutes to train. For the experiments with k=5, we use 10^{-5} as the learning rate.

For the point cloud experiments, we use a batch size of 32 and train for 20 epochs, which takes about four hours. We train Pure and Diag with the same batch sizes and epochs. The training of the Diag baseline takes four hours as well, and Pure is trained within three hours. When solving a QUBO on a QA, we anneal 100 times and pick the lowest-energy solution. We



(a) Chimera (b) Pegasus

Figure 6.5: Visualisation of qubit connectivities. (a) The connectivity pattern of the physical qubits in the Chimera architecture. The unit cells (green boxes) have fewer interconnections than on Pegasus. (b) The connectivity pattern of the physical qubits in the Pegasus architecture. Green, red, and yellow correspond to one problem instance each. Images due to D-Wave [73].

access the QA via Leap 2 [71] using the Ocean SDK [77]. When solving with SA, we use 100 iterations from the default `neal` SA sampler.

6.9 Graph Matching

6.9.1 Problem Description

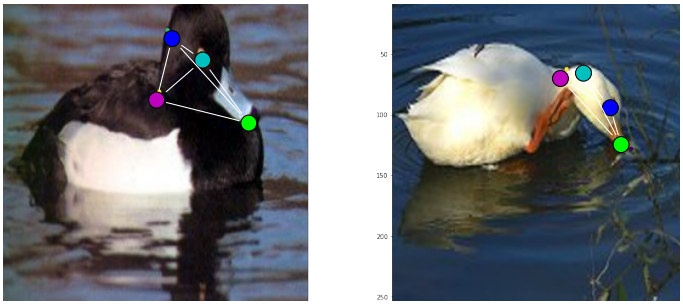
Here, we describe the design of the problem description $\mathbf{p}$ for graph matching. We use $\mathbf{p} = \text{vec}(\mathbf{W})$, where the diagonal of $\mathbf{W}$ contains cosine similarities between the feature vectors extracted with AlexNet [154] pre-trained on ImageNet [88] of all pairs of key points. The off-diagonal follows the geometric term described in (Eq. (7)) from Torresani *et al.* [256]. In particular, we use the term $\mathbf{W}_{\text{geom}}$ from Eq. (7) from Torresani *et al.* [256] with minus signs in the beginning and in the exponential, and set η=0.98. The convex combination with $\mathbf{W}_{\text{Alex}}$, where the cosine similarities of the feature vectors are on the diagonal, is then:

$$\mathbf{W} = \tau \mathbf{W}_{\text{Alex}} + (1 - \tau)\mathbf{W}_{\text{geom}}. \tag{6.9}$$

We choose τ=0.81 and η=0.98 such that the QAP often coincides with the ground-truth correspondences (see Table 6.1b, "Direct" from the main paper).

6.9.2 Failure Case

We show a failure case of our method when applied to graph matching in Fig. 6.6. It occurs due to large differences in the observed appearance.



(a) Source (b) Matching

Figure 6.6: Failure case for graph matching. We visualise the ground truth of an image pair. Here, our method does not find the correct matching: Only the beak and neck are matched correctly, while the geometric information for the other key points differs too strongly.

Table 6.9: Average success probabilities of different QGM variants [237] over 141 problem instances on RandGraph compared to QuAnt with $k = 5$, in %.

Inserted	Baseline	Row-wise	Ours
0.22	0.07	0.07	**26**

6.10 Detailed Comparison with QGM

Here, we compare our method with QGM [237] in detail. Note that the focus of both works differs. Their work focuses more on the probability distribution of the retrieved solutions. Our work is more concerned with incorporating the quantum annealer into the training pipeline. When training the neural network, L_{gap} (6.1) uses the retrieved solution with the smallest energy across anneals, while they are also interested in the success probabilities, *i.e.,* the probability to get the best solution across anneals.

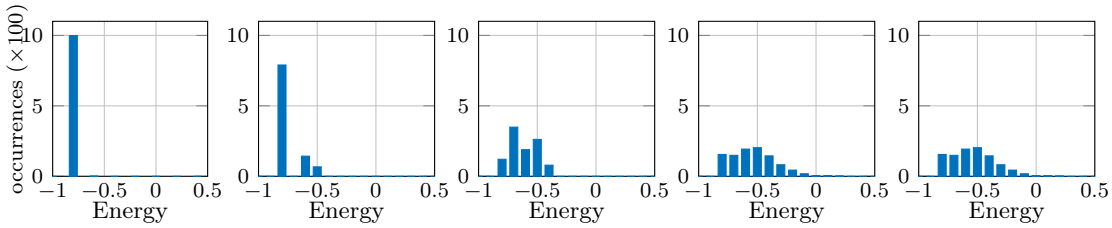
The *individual* QUBOs occurring in our QuAnt framework are much easier to solve by the QA than QUBOs that would arise in QGM [237]. To show this, we compute the average success probabilities of the various methods from QGM [237] over 141 instances of RandGraph with k=5; see Table 6.9. We also apply QuAnt to these problem instances. We solve the resulting QUBO with QA and find the average probability to be 26% with a standard deviation of 18%, better than any method from the QGM paper [237].

This difference is not surprising since we construct our method such that we only use trivial embeddings and do not need to apply the `minorminer` heuristic [52]. Because of that, for RandGraph with $k = 5$, our method needs only 15 physical qubits while their baseline and row-wise methods need 89 qubits, on average, and a chain length of 4; their *Inserted* method needs, on average, 39 qubits and a chain length of 3 on D-Wave Advantage. Note that a heuristic search for better penalty parameters, as in Q-Sync [40], could give rise to better results for the methods from QGM [237] in Table 6.9. However, the corresponding embeddings would still be problematic. Directly using the binary encoding for permutations [107] requires additional qubits because the problem would a priori not be quadratic.

6.11 Solution Quality of SA and QA

In the main paper, we show that training with QA yields better performance than training with SA. Here, we analyse the quality of the solutions found by both techniques further. Fig. 6.7 contains histograms that depict the output of the quantum annealer and the two different simulated annealing solvers from `neal` [79] and from `dimod` [78]. In contrast to the solver from `dimod`, `neal` is highly optimised for performance, so we used it for our experiments.

We focus our analysis on the number of sweeps in SA, *i.e.,* the number of steps in the 'cooling' schedule. We observe that it strongly influences the quality of the second-best solution.



(a) SA, 10^3 sweeps (b) SA, 10 sweeps (c) SA, 2 sweeps (d) `Dimod` (e) QA

Figure 6.7: Energy histograms of (ideally optimal) 10^3 samples for SA (different number of sweeps and dimod) and QA on one instance of RandGraph with k=5 after 450 epochs training.

Table 6.10 illustrates this by averaging the fraction of the second-best energies over 141 instances and analysing 1000 samples from different solvers. We see that the quantum annealer produces the second-best samples with the lowest energies.

Note, however, that we do not claim that this is an intrinsic general advantage of QA over SA, but merely that in our setting, QA outperforms SA. Still, prior work [268] also reaches the conclusion that quantum annealing has much potential for finding reasonable near-optimal solutions.

The `dimod` sampler also produces second-best solutions with low energies but is computationally expensive [78]. This is, perhaps, because many non-optimal solutions are produced compared to the implementation from `neal`.

Table 6.10: Second-best energies of SA relative (in %) to the second-best energies of QA. We report the mean and std. deviation over 141 instances. The higher the better.

SA (`neal`), 10^3 sweeps	SA (`neal`), 10 sweeps	SA (`dimod`)
94.2± 10.3	86.0 ± 19	99.8 ± 0.9

6.12 3D Rotation Estimation

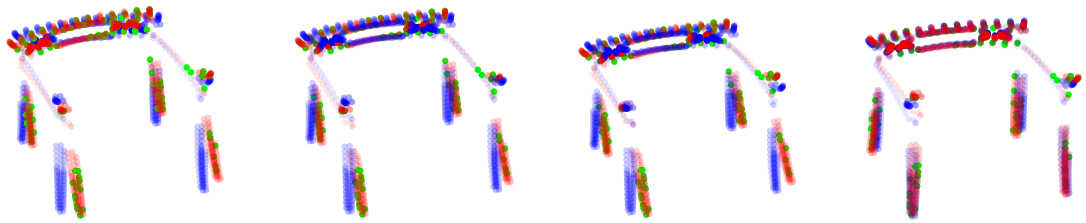
6.12.1 Three Stages (Euler Angles)

We obtain improved results when regressing three Euler angles one after another compared to direct regression of three angles. Thus, we use one stage per angle, *i.e.,* with one network per stage. The training setup is as follows. We feed the first network with problem instances where

Table 6.11: Comparison to general baselines on rotation estimation. We report the mean of the per-experiment *median* and std. deviation across experiments for three different random seeds.

	Ours	Diag	Pure
L=3, H=32	6.0 ± 3.6	**5.3 ± 1.1**	7.0 ± 1.0
L=3, H=78	**4.0 ± 1.0**	5.0 ± 0.0	7.0 ± 0.0
L=5, H=32	**3.7 ± 1.2**	5.0 ± 0.0	16.3 ± 7.2
L=5, H=78	**3.7 ± 0.6**	5.0 ± 0.0	9.0 ± 1.0

$\alpha, \beta, \gamma \neq 0$. The network then regresses α. We feed the second stage network with problem instances, where $\alpha = 0$ and $\beta, \gamma \neq 0$. Subsequently, the network regresses β. Lastly, we feed the third network with problem instances, where $\alpha, \beta = 0$ and $\gamma \neq 0$. Here it regresses γ. During test time, the first network determines α, which is then applied to the input. The updated input is re-encoded before the second stage, which outputs β. Finally, with α and β applied already, the third stage regresses γ. Fig. 6.8 shows how the three different networks regress the angles and how the solution progresses towards the final one.



(a) Initial problem instance (b) Application of the α rotation (c) Application of the α, β rotations (d) Application of the α, β, γ rotations

Figure 6.8: Visualisation of the different steps of our rotation-estimation network. We show (blue) the original 3D point cloud and (red) the rotated point cloud. The green points are points of the red point cloud with unknown correspondences. Here, 10% of the correspondences are unknown.

6.12.2 Variance across Runs

To better judge the stability under different random seeds, we repeat the main experiment from the paper three times for our QuAnt method and each baseline. In Table 6.11, we report the mean and std. deviation of the median. Here, similar to the results from the main paper, we outperform the Diag and Pure baselines in all but one setting.

6.12.3 Training on Noisy Data

Table 6.12 shows how our method performs on noise-free and noisy test data after training on noisy data. We observe that the noisy training data appears to negatively affect the training and its performance drops, while Diag improves and Pure remains unchanged.

Table 6.12: Robustness to varying amounts of incorrect test-time correspondences in rotation estimation. We report the mean/median error for L=3, H=32. The first column specifies the percentage of incorrect correspondences at test time.

(a) Training without noise

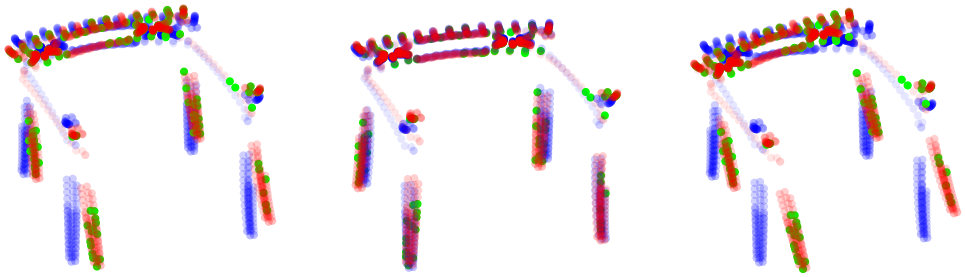
%	**Ours**	Procrustes	Diag	Pure
0	3.9 / 4.0	**0.0** / **0.0**	5.6 / 6.0	8.1 / 8.0
1	**3.4** / **3.0**	5.8 / 3.0	5.7 / 6.0	8.2 / 8.0
5	**3.4** / **3.0**	25.7 / 13.0	6.0 / 6.0	8.2 / 8.0
10	**3.2** / **3.0**	43.8 / 21.0	6.2 / 6.0	8.2 / 8.0
15	**3.5** / **3.0**	64.7 / 58.0	6.2 / 6.0	8.2 / 8.0
20	**3.7** / **3.0**	75.3 / 79.0	5.8 / 6.0	8.2 / 8.0

(b) Training with 10% incorrect correspondences

%	**Ours**	Procrustes	Diag	Pure
0	4.4 / 4.0	**0.0** / **0.0**	4.6 / 5.0	8.1 / 8.0
1	**4.3** / **4.0**	5.8 / 3.0	5.1 / 5.0	8.3 / 8.0
5	**5.0** / **5.0**	25.7 / 13.0	5.1 / 5.0	8.2 / 8.0
10	**5.2** / **5.0**	43.8 / 21.0	5.2 / 5.0	8.2 / 8.0
15	5.2 / 5.0	64.7 / 58.0	**5.0** / **5.0**	8.2 / 8.0
20	5.6 / 6.0	75.3 / 79.0	**4.9** / **5.0**	8.2 / 8.0

6.12.4 Qualitative Comparison on Noisy Test Data

Fig. 6.9 visualises differences between our solution after training on noise-free data and Procrustes alignment on a problem with noisy data (unknown correspondences).



(a) Initial problem setting (b) Our solution (c) Solution by Procrustes

Figure 6.9: Comparison of initial input rotation, our method and Procrustes. The initial 3D point cloud is blue and the rotated one is red, where the unknown correspondences are displayed in green. Here, 10% of the correspondences are unknown.

6.12.5 Comparison to AQM [111]

QuAnt can estimate 3D rotations with known point matches. However, AQM [111] would require 81 densely connected logical qubits, which is not supported by the current quantum-hardware generations. Hence, we cannot compare against AQM for this problem setting and instead only compare to classical methods such as Procrustes.

6.13 2D Point Set Registration

6.13.1 Setting Details

We encode the point set registration instances similar to Golyanik *et al.* [111]. As the correspondences between the template and the reference in point set registration are not known, we use k-nearest neighbours to find possible correspondences. Our network is trained with three nearest neighbours per each template point.

Table 6.13: Comparison of QuAnt to general baselines on point set registration. We report the mean of the per-experiment *median* and std. deviation across experiments for three different random seeds.

	Ours	Diag	Pure
L=3, H=32	**4.8 ± 0.6**	6.8 ± 0.2	5.8 ± 0.9
L=3, H=78	**3.7 ± 0.3**	5.1 ± 0.4	4.8 ± 0.3
L=5, H=32	**4.6 ± 0.1**	7.1 ± 1.0	7.1 ± 1.6
L=5, H=78	**3.4 ± 0.1**	4.9 ± 0.0	7.9 ± 2.7

Table 6.14: Robustness to varying amounts of uniform noise. We report the mean/median error for L=3, H=32. The first column specifies the range of the added uniform noise, in %, of the maximum extent of the point cloud.

%	**Ours**	Diag	Pure	AQM
0	***7.4 / 4.2***	11.3 / 7.0	7.8 / 6.6	**4.3** / **2.6**
5	***6.7 / 3.8***	11.7 / 7.3	8.0 / 6.8	**4.5** / **2.9**
10	***7.2 / 4.5***	12.2 / 6.8	8.6 / 7.0	**5.6** / **3.8**
15	***8.2 / 4.9***	12.6 / 6.8	9.7 / 8.0	**5.6** / **3.8**
20	***11.0 / 6.0***	13.9 / 8.2	14.3 / 10.4	**5.9** / **3.3**

6.13.2 Variance

In Table 6.13, we report the mean median. Here, we outperform the baselines in all cases. In nearly all setups, we outperform the baselines, and only for one case, we are on par with the Pure baseline. Still, even in that case, QuAnt performs more consistently, as evidenced by its lower std. deviation. These experiments show that we consistently outperform the baselines and the performance is not dependent on the random seed.

6.13.3 Noise

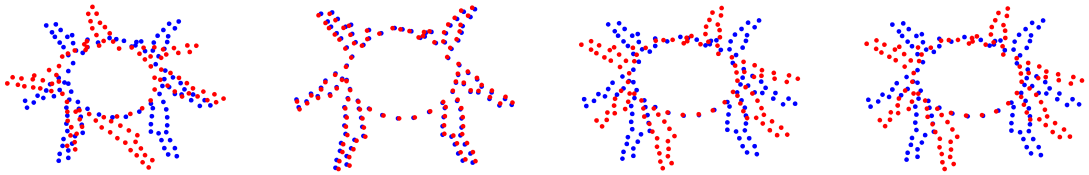
In addition to the experiments in the main paper that uses the largest architecture, we also test the noise resistance on the smallest network setup with $L = 3$, $H = 32$; see Table 6.14. Here, while QuAnt is better than the baselines, we do not outperform AQM. However, *by construction, AQM is an upper bound for our method as the matrix introduced by Golyanik et al. [111] is the same as our input into the network, but it gets directly solved by the QA.*

6.13.4 Qualitative Ablation Results

In addition to the quantitative loss ablation in the paper, we visualise the effect of the losses here. In Figure 6.10, the full loss results in a nearly ground-truth rotation. However, if we leave out L_{gap} or L_{MLP} during training, a significant reduction in rotational accuracy is visible.

6.13.5 Failure Case

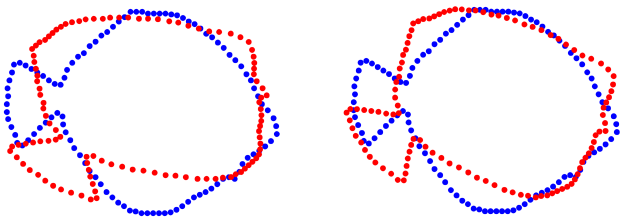
We continue our analysis with failure cases. Results in the main paper show that an increasing input angle leads to a reduction in the accuracy of our regressed angle. This can be traced back to our problem-instance encoding as Golyanik *et al.* [111] mention that an increasing angle makes finding the correspondences more error-prone. Therefore, an imperfect input encoding makes it is also more likely for us to regress wrong angles.



(a) Initial problem instance (b) Ours with the full loss (c) Ours without L_{MLP} (d) Ours without L_{unique}

Figure 6.10: Qualitative loss ablations. We show the original point cloud (blue) and rotated point cloud (red). Removing either L_{unique} or L_{mlp} leads to significantly worse results.

Similar to most prior work on point set registration, nearly symmetric shapes can be difficult, as most points can be nearly perfectly aligned even with wrong rotations. Fig. 6.11 contains such a failure case. The initial angle of this problem instance, 50.8°, is relatively large for our setup, and the shape (which looks like the silhouette of a fish) is nearly rotationally symmetric. In cases like this, our method has difficulties regressing the correct rotation after a single QUBO sampling.



(a) Initial problem instance (b) Failing to regress correct rotation

Figure 6.11: Example of a failure case in point set registration. We show the initial image (blue) as well as the rotated image (red).

Chapter 7

Quantum Motion Segmentation

7.1 Contextualization

In the following parts of the work "Quantum Motion Segmentation", published at ECCV 2022, are presented. While the application setting differs somewhat from previous publications, the core pipeline for applying quantum annealing remains largely the same. However, particularly interesting is that there are also iterative methods using quantum annealing in a divide-and-conquer algorithm present.

Vladislav Golyanik, Frederica Arrigoni and Elisa Ricci came up with the idea. Willi Menapace, Vladislav Golyanik and Frederica Arrigoni performed the experiments. Marcel Seelbach Benkner contributed by completing the proof of proposition 1 in the publication. All people contributed by writing. Note that this publication is only listed here for completeness and, all in all, my contributions were smaller compared with the other works presented in this thesis.

7.2 Introduction

Motion segmentation is a challenging problem that seeks to identify independent motions in two or several input images. In the paper we introduced the first algorithm for motion segmentation that relies on adiabatic quantum optimization of the objective function.

7.3 Background

Our work is inspired by Arrigoni and Pajdla [19] (reviewed in Sec. 7.3.1), where a convenient matrix representation is introduced for motion segmentation from pairwise correspondences. In Sec. 7.4 we will show how to rewrite such framework in terms of a QUBO, in order to enable adiabatic quantum optimization.

7.3.1 Motion Segmentation

The objective of the motion segmentation problem is to group key-points in multiple images according to a number of motions. We use the following notation: n is the number of images;

p_i is the number of key-points in image i; $p = \sum_{i=1}^{n} p_i$ is the total amount of key-points; d is the number of motions (known by assumption). We focus here on motion segmentation from pairwise correspondences [20, 19], which can be addressed in two steps: 1) motion segmentation is addressed on different *pairs* of images independently, which in turn can be done via multi-model fitting [171, 172, 26]; 2) the results derived in the first step are globally combined, thus producing the required *multi-frame* segmentation.

As shown in [19], motion segmentation can be seen as a "synchronization" of binary matrices. Indeed, the result of motion segmentation in two images i and j can be represented as a matrix $Z_{ij} \in \{0,1\}^{p_i \times p_j}$ as follows:

- $[Z_{ij}]_{h,k} = 1$ if point h in image i and point k in image j belong to the same motion;
- $[Z_{ij}]_{h,k} = 0$ otherwise.

The (known) binary matrix Z_{ij} is referred to as the *partial segmentation* or *relative segmentation* of the pair (i, j). It is a *local* representation of segmentation, since it reveals which points in two different images belong to the same motion, but it does not reveal which motion it is with respect to other pairs. Similarly, our desired output can be represented as a matrix $X_i \in \{0,1\}^{p_i \times d}$ as follows:

- $[X_i]_{h,k} = 1$ if point h in image i belongs to motion k;
- $[X_i]_{h,k} = 0$ otherwise.

The binary matrix X_i is called the *total segmentation* or *absolute segmentation* of image i. Observe that the number of rows is equal to the number of points while the number of columns is equal to the number of motions. Note also that it is a *global* representation of segmentation since it reveals the membership of all points with respect to an absolute order of motions. The notions of absolute and relative segmentations are illustrated in Fig. 7.2.

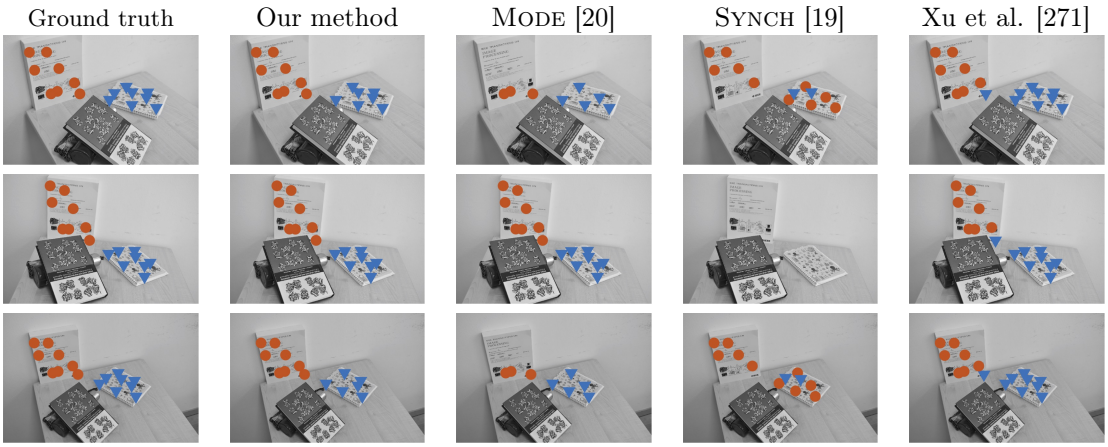


Figure 7.1: Qualitative results on sample images from the new Q-MSEG dataset, where each color (symbol) represents a distinct planar motion. On average, the accuracies of our QUMOSEG-V1, MODE [20], SYNCH [19] and Xu et al. [271] are 0.97, 0.93, 0.93 and 0.89, respectively, on problems with 96 qubits. In the shown example, our approach outperforms the competitors.

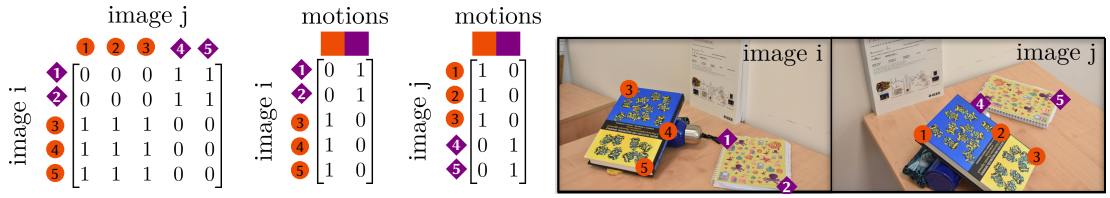


Figure 7.2: Matrix representation of motion segmentation.

Remark 7.1. Let $\mathbf{m}_i$ be a vector of length d such that $[\mathbf{m}_i]_h$ counts how many points in image i belongs to motion h. Then the columns in X_i sum to $\mathbf{m}_i$:

$$\mathbf{1}_{p_i}^T X_i = \mathbf{m}_i^T, \tag{7.1}$$

where $\mathbf{1}$ is a vector of ones (with length given as subscript). Note also that the product $X_i^T X_i$ is a $d \times d$ diagonal matrix:

$$X_i^T X_i = \mathrm{diag}(\mathbf{m}_i). \tag{7.2}$$

These simple properties will be exploited later.

The connection between relative and absolute segmentations [19] is given by:

$$Z_{ij} = X_i X_j^{\mathsf{T}}. \tag{7.3}$$

Recall that the left side in the above equation is known whereas the right side is unknown. In general, there are multiple image pairs giving rise to an equation of the form (7.3), which can be conveniently represented as the edge set $\mathcal{E}$ of a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{1, \ldots, n\}$ denotes the vertex set. In other terms, each vertex represents an image and an edge is present between two vertices if and only if the relative segmentation of that image pair is available. Thus motion segmentation can be cast to the task of recovering $X_1, \ldots, X_n$ starting from Z_{ij} with $(i, j) \in \mathcal{E}$, such that (7.3) is satisfied. This is also called *synchronization* [17].

7.4 Our Quantum Approach

At the core of our approach is a QUBO formulation of motion segmentation. This permits—for the first time in the literature—to solve the segmentation task via quantum optimization. Bringing it into a QUBO form is not straightforward and requires more analytical steps compared to previous work on quantum synchronization [40], as will be clarified later. We propose two methods: the first one is based on a dense matrix, and, hence, can require an increased embedding size (Sec. 7.4.1); the second one, instead, is based on a sparse matrix but relies on additional assumptions (Sec. 7.4.2). Both variants are principled (*i.e.,* are equivalent to synchronization) and can solve real-world problems (see Sec. 5.6).

7.4.1 QuMoSeg-v1

As explained in Sec. 7.3.1, motion segmentation can be posed as computing absolute segmentations (*i.e.,* $X_1, \dots, X_n$) starting from pairwise segmentations Z_{ij} with $(i,j) \in \mathcal{E}$ such that $Z_{ij} = X_i X_j^T$. In the presence of noise, the task is to solve the following optimization problem:

$$\begin{aligned} &\min_{X_1,\dots X_n} \sum_{(i,j)\in\mathcal{E}} ||Z_{ij} - X_i X_j^\intercal||_F^2, \\ &\text{s.t. } \operatorname{vec}(X_i) \in \mathcal{B}^{p_i}, \quad X_i \mathbf{1}_d = \mathbf{1}_{pi} \quad \forall i = 1, \dots, n, \end{aligned} \tag{7.4}$$

where $\mathcal{B}^k$ denotes the set of binary vectors of length k, and $\operatorname{vec}(\cdot)$ denotes the vectorization operator that transforms a matrix into a vector by stacking the columns one under the other. Recall that X_i has size $p_i \times d$, so $\operatorname{vec}(X_i)$ has length dp_i and should be a binary vector. The constraint $X_i \mathbf{1}_d = \mathbf{1}_{pi}$ means that each row sums to 1. Indeed, each row in X_i has exactly one entry equal to 1 (whereas all other entries are zero), meaning that each point should belong to exactly one motion. The cost in (7.4) measures, for each image pair, the discrepancy (in the Frobenius norm sense) between the input relative segmentation (*i.e.,* Z_{ij}) and the relative segmentation derived from the sought absolute segmentations (*i.e.,* $X_i X_j^\intercal$). It is also known as the *consistency error* [17].

Proposition 7.2. *Problem* (7.4) *is equivalent to*

$$\begin{aligned} &\max_{X_1,\dots X_n} \sum_{(i,j)\in\mathcal{E}} \operatorname{Tr}(X_i^T (2Z_{ij} - \mathbf{1}_{p_i \times p_j}) X_j) \\ &s.t. \ \operatorname{vec}(X_i) \in \mathcal{B}^{p_i}, \quad X_i \mathbf{1}_d = \mathbf{1}_{pi} \quad \forall i = 1, \dots, n \end{aligned} \tag{7.5}$$

where $\mathbf{1}_{p_i \times p_j}$ *denotes a* $p_i \times p_j$ *matrix of ones.*

Proof. For simplicity of exposition, we drop constraints and focus on the cost function itself. By computation and exploiting (7.2), we obtain:

$$\begin{aligned} ||Z_{ij} - X_i X_j^\intercal||_F^2 = \operatorname{Tr}(Z_{ij}^\intercal Z_{ij}) + \operatorname{Tr}(X_j X_i^T X_i X_j^T) - 2\operatorname{Tr}(Z_{ij}^T X_i X_j^T) = \\ \operatorname{Tr}(Z_{ij}^\intercal Z_{ij}) + \operatorname{Tr}(\operatorname{diag}(\mathbf{m}_i)\operatorname{diag}(\mathbf{m}_j)) - 2\operatorname{Tr}(X_i^T Z_{ij} X_j) = \\ = \operatorname{Tr}(Z_{ij}^\intercal Z_{ij}) + \mathbf{m}_i^T \mathbf{m}_j - 2\operatorname{Tr}(X_i^T Z_{ij} X_j). \end{aligned} \tag{7.6}$$

Note that the first term is constant, for it depends on the input Z_{ij} only, hence it can be ignored in the optimization. As for the second term, using (7.1), we get:

$$\mathbf{m}_i^T \mathbf{m}_j = \operatorname{Tr}(\mathbf{m}_i \mathbf{m}_j^T) = \operatorname{Tr}(X_i^T \mathbf{1}_{p_i} \mathbf{1}_{p_j}^T X_j) = \operatorname{Tr}(X_i^T \mathbf{1}_{p_i \times p_j} X_j). \tag{7.7}$$

Hence, the optimization in (7.4) is equivalent to (7.5).

□

We now rewrite (7.5) in a compact form via the following notation, where all the mea-

sures/unknowns are grouped into block-matrices $X \in \mathbb{R}^{p\times d}$ and $Z \in \mathbb{R}^{p\times p}$:

$$X = \begin{bmatrix} X_1 \\ X_2 \\ \dots \\ X_n \end{bmatrix}, \quad Z = \begin{bmatrix} 0 & Z_{12} & \dots & Z_{1n} \\ Z_{21} & 0 & \dots & Z_{2n} \\ \dots & & & \dots \\ Z_{n1} & Z_{n2} & \dots & 0 \end{bmatrix}. \tag{7.8}$$

Proposition 7.3. *Problem* (7.4) *is equivalent to*

$$\begin{gathered} \max_X \operatorname{vec}(X)^T (I_{d\times d} \otimes (2Z - \mathbf{1}_{p\times p}))\operatorname{vec}(X), \\ s.t.\ \operatorname{vec}(X) \in \mathcal{B}^{dp}, \quad X\mathbf{1}_d = \mathbf{1}_p. \end{gathered} \tag{7.9}$$

Proof. Let us define $W = 2Z - \mathbf{1}_{p\times p}$. Using (7.8), we get:

$$\sum_{(i,j)\in\mathcal{E}} \operatorname{Tr}(X_i^T(2Z_{ij} - \mathbf{1}_{p_i\times p_j})X_j) = \operatorname{Tr}(X^T W X). \tag{7.10}$$

The above equation can be further simplified by exploiting properties of the trace operator[1] and the Kronecker product[2] denoted by $\otimes$, resulting in:

$$\operatorname{Tr}(X^T W X) = \operatorname{vec}(X)^T \operatorname{vec}(WX) = \operatorname{vec}(X)^T (I_{d\times d} \otimes W)\operatorname{vec}(X), \tag{7.11}$$

where $I_{d\times d}$ denotes the $d \times d$ identity matrix. Hence, the objective function (7.9) is the same as (7.5), which in turn is equivalent to (7.4), as shown in Prop. 7.2. As for constraints, it is easy to see that $\operatorname{vec}(X_i) \in \mathcal{B}^{p_i}$ and $X_i\mathbf{1}_d = \mathbf{1}_{pi}$ translate into $\operatorname{vec}(X) \in \mathcal{B}^{dp}$ and $X\mathbf{1}_d = \mathbf{1}_p$, when considering all the unknowns simultaneously as stored in the block-matrix X. Hence we get the thesis. □

Corollary 1. Problem (7.4) can be mapped into a QUBO problem

$$\min_{\mathbf{y}\in\mathcal{B}^k} \mathbf{y}^\mathsf{T} Q\mathbf{y} + \lambda_1 ||A\mathbf{y} - \mathbf{b}||^2, \tag{7.12}$$

where $Q = -I_{d\times d} \otimes (2Z - \mathbf{1}_{p\times p})$, $\mathbf{y} = \operatorname{vec}(X)$, $A = (\mathbf{1}_d^T \otimes I_{p\times p})$, $\mathbf{b} = \mathbf{1}_p$.

Proof. Problem (7.9) is the maximization of a quadratic cost function with binary variables and linear constraints, hence definitions of Q and $\mathbf{y}$ are immediate. The linear constraints $X\mathbf{1}_d = \mathbf{1}_p$ can be easily mapped into the canonical form $A\mathbf{y} = \mathbf{b}$ via properties of Kronecker product[2] and vectorization. □

Remark 7.4. $Q = -I_{d\times d} \otimes (2Z - \mathbf{1}_{p\times p})$ is symmetric and its size is $dp \times dp$, where d is the number of motions and p is the total amount of points over all images. Note also that the size of the optimization variable $\mathbf{y}$ is dp, so it scales linearly with the number of points (assuming $d \ll p$, which is usually the case in practice).

[1] For any matrices A, B of proper dimensions we have: $\operatorname{Tr}(A^T B) = \operatorname{vec}(A)^T \operatorname{vec}(B)$.

[2] For any matrices A, B, Y of proper dimensions, the Kronecker product [166] satisfies: $\operatorname{vec}(AYB) = (B^T \otimes A)\operatorname{vec}(Y)$.

To summarize, the synchronization formulation for motion segmentation can be cast to a QUBO, thus enabling adiabatic quantum optimization. This gives rise to the first variant of our approach, which is called QuMoSeg-v1.

7.4.2 QuMoSeg-v2

Note that the block-matrix Z storing all partial segmentations is *sparse*, *i.e.,* most of its entries are zero (see Fig. 7.2 for an example of a partial segmentation). However, the matrix $2Z - \mathbf{1}_{p\times p}$ (which appears in the definition of Q in (7.12)) is *dense* (it has only -1 or +1 as possible entries). This can result in an increased size of the minor embedding on an AQC, which is undesirable in practice. This observation motivates the need for an alternative method based on a sparse matrix, which is explored here. This comes at the price of having additional assumptions, as shown below.

Proposition 7.5. *Let us assume that the amount of points per motion is known in each image, namely* $\mathbf{m}_i$ *is known* $\forall i = 1, \dots, n$*, where the d-length vector* $\mathbf{m}_i$ *is defined in Remark 7.1. Then, Problem* (7.4) *is equivalent to*

$$\begin{aligned} &\max_{X_1,\dots X_n} \sum_{(i,j)\in\mathcal{E}} \operatorname{Tr}(X_i^T Z_{ij} X_j), \\ s.t.\ \ &\operatorname{vec}(X_i) \in \mathcal{B}^{p_i}, \quad X_i \mathbf{1}_d = \mathbf{1}_{pi}, \quad \mathbf{1}_{p_i}^T X_i = \mathbf{m}_i^T \quad \forall i = 1, \dots, n. \end{aligned} \tag{7.13}$$

Proof. The starting point is Eq. (7.6), which is copied here as a reference:

$$||Z_{ij} - X_i X_j^\intercal||_F^2 = \operatorname{Tr}(Z_{ij}^\intercal Z_{ij}) + \mathbf{m}_i^T \mathbf{m}_j - 2\operatorname{Tr}(X_i^T Z_{ij} X_j). \tag{7.14}$$

As already observed, the first term is constant, for it depends on Z_{ij} only; hence, it can be ignored in the optimization. Also the second term is constant, since $\mathbf{m}_i$ is known for each image i *by assumption*. Thus, the objective function (7.4) is equivalent to (7.13). Observe that we should add extra constraints to take into account our additional assumptions, which force the amount of points per motion to be equal to some predefined values in every image, namely $\mathbf{1}_{p_i}^T X_i = \mathbf{m}_i^T$ (see also Remark 7.1). Hence we get the thesis. □

Remark 7.6. Note that the knowledge of the number of points per motion was indeed essential in the proof of Prop. 7.5. In other terms, the synchronization problem (7.4) **is not equivalent** to (7.13) without such an assumption.

Proposition 7.7. *If the amount of points per motion is known for each image, then Problem* (7.4) *is equivalent to*

$$\begin{aligned} &\max_X \operatorname{vec}(X)^T (I_{d\times d} \otimes Z) \operatorname{vec}(X), \\ s.t.\ \ &\operatorname{vec}(X) \in \mathcal{B}^{dp}, \quad X\mathbf{1}_d = \mathbf{1}_p, \quad KX = M, \end{aligned} \tag{7.15}$$

where K *and* M *are defined as follows:*

$$K = \operatorname{diag}(\mathbf{1}_{p_1}^T, \mathbf{1}_{p_2}^T, \dots, \mathbf{1}_{p_n}^T), \quad M = \begin{bmatrix} \mathbf{m}_1^T & \mathbf{m}_2^T & \dots & \mathbf{m}_n^T \end{bmatrix}^T. \tag{7.16}$$

Proof. Problem (7.13) can be easily turned into the form (7.15) following the same reasoning as in the proof of Prop. 7.3. Concerning the additional constraints, it is easy to see that $KX = M$ is just a compact way of storing all the equations of the form $\mathbf{1}_{p_i}^T X_i = \mathbf{m}_i^T$ simultaneously. □

Corollary 2. If the amount of points per motion is known for each image, then Problem (7.4) can be mapped into a QUBO problem, namely:

$$\min_{\mathbf{y} \in \mathcal{B}^{dp}} \mathbf{y}^\mathsf{T} P \mathbf{y} + \lambda_2 ||A\mathbf{y} - \mathbf{b}||^2 + \lambda_3 ||E\mathbf{y} - \mathbf{f}||^2, \tag{7.17}$$

where $P = -I_{d\times d} \otimes Z,\ \mathbf{y} = \text{vec}(X),\ A = (\mathbf{1}_d^T \otimes I_{p\times p}),\ \mathbf{b} = \mathbf{1}_p,\ \mathbf{f} = \text{vec}(M)$, and $E = (I_{d\times d} \otimes K)$.

Proof. Problem (7.15) is the maximization of a quadratic cost with binary variables and linear constraints, hence definitions of P and $\mathbf{y}$ are immediate. The linear constraints $X\mathbf{1}_d = \mathbf{1}_p$ and $KX = M$ can be mapped into the forms $A\mathbf{y} = \mathbf{b}$ and $E\mathbf{y} = \mathbf{f}$ via properties of Kronecker product[2] and vectorization. □

In summary, we derived an alternative QUBO formulation for motion segmentation with a *sparse* matrix: P matrix defined in (7.17) is sparse as it inherits the same sparsity pattern as Z. This was possible **under simplified assumptions**, *i.e.,* the knowledge of the amount of points per motion in all images. This gives rise to the second variant of our approach, which is called QuMoSeg-v2.

7.5 Conclusion

This work proposed the first motion segmentation approach for an adiabatic quantum computer, which shows highly promising results and reaches state-of-the-art accuracy on a wide range of problems. While we did not present a detailed experimental analysis here, one can find more results in the original publication. Some qualitive results are shown here in 7.1.

Chapter 8

Compensating Connectivity Restrictions in Quantum Annealers via Splitting and Linearization Techniques

8.1 Contextualization

This work is currently under review. It presents a new iterative method for solving QUBOs that works with a hardware-restricted QUBO solver at each step. The core idea is to embed the couplings from the problem graph trivially onto the hardware and, in each iteration, linearize parts that are not supported by the hardware. Additionally, a damping term is included, so that the previous iterate still influences how well the initial QUBO is approximated.

A proof of weak monotonicity is presented, and the method is extensively benchmarked on the Max-cut dataset MQLib [95].
Marcel Seelbach Benkner performed the experiments and theoretically explored the relation to other optimization methods in the literature. Michael Möller came up with the idea and the convergence proof for the discrete case. He also delivered insights about related continuous optimization methods. Martin Kliesch influenced the experimental design, shared expertise on the physics side and proposed MQlib as a benchmark dataset. Zorah Lähner and Vladislav Golyanik improved experimental design through active and helpful discussion.

8.2 Abstract

Current quantum annealing experiments often suffer from restrictions in connectivity in the sense that only certain qubits can be coupled to each other. The most common strategy to overcome connectivity restrictions so far is by combining multiple physical qubits into a logical qubit with higher connectivity, which is achieved by adding terms to the Hamiltonian. Practically, this strategy is implemented by finding a so-called minor embedding, which is in itself an NP-hard

problem. In this work, we present an iterative algorithm that does not need additional qubits but instead efficiently uses the available connectivity for different parts of the problem graph in every step. We present a weak monotonicity proof and benchmark our algorithm against the default minor-embedding algorithm on the D-Wave quantum annealer and multiple simple local search variants. While most of the experiments to compare the different iterative methods are performed with simulated annealing solvers, we also confirm the practicality of our method with experiments on the D-Wave Advantage quantum annealer.

8.3 Introduction

Quantum annealing is a recently emerging optimization method that promises to outperform, e.g., simulated annealing in rugged energy landscapes. The term quantum annealing is often used to refer to the current experiments motivated by adiabatic quantum computing, which may not achieve perfect adiabaticity [114]. Due to this imperfection, the systems are not necessarily in the ground state throughout the whole evolution, but also go to low-energy excited states. While adiabatic quantum computing is under some theoretical assumptions polynomially equivalent to gate-based quantum computing [10], we are in this work mostly interested in quantum annealing as a heuristic solver for quadratic unconstrained optimization (QUBO) problems. These problems have the form

$$\hat{x} := \underset{x \in \{0,1\}^n}{\arg\min}\, x^T Q x, \tag{8.1}$$

where $Q \in \mathbb{R}^{n \times n}$ is a matrix of couplings, which we assume to be symmetric without restriction of generality. For QUBOs, there is also an equivalent formulation to the Ising problem

$$\hat{s} := \underset{s \in \{-1,1\}^n}{\arg\min}\, s^T A s + s^T b, \tag{8.2}$$

with $A = \frac{1}{4}Q$ and $b = \frac{1}{2}Q\mathbf{1}$, where $\mathbf{1} \in \mathbb{R}^n$ is a vector with only ones as entries. QUBO problems can be solved with traditional algorithms, like simulated annealing, but the energy landscape can become very complicated, and the problem of finding the optimum is NP-hard in general. It is believed that quantum annealing can often handle these landscapes more efficiently and robustly [81, 86], and there are even problems where a quadratic speedup compared to the best foreseeable classical algorithm is mathematically proven [223].

To solve the minimization (8.1) with quantum annealing in a straightforward way, n logical qubits are necessary. However, most current quantum annealing experiments are still quite restricted in the number of qubits and thus in the problem size. The largest system is currently manufactured by the company D-Wave[1] and uses superconducting flux qubits [182]. Although the D-Wave Advantage machine has over 5000 qubits, it can only solve QUBO problems up to the size of ≈ 150 variables if Q is a dense matrix [5]. This overhead is due to physical limitations in connecting qubits: a non-zero entry in Q requires a connection between the two corresponding qubits, but it is currently not possible to manufacture all qubits such that they are all connected. To overcome this limitation, multiple physical qubits are combined into a logical qubit with

[1] https://www.dwavesys.com/

higher connectivity, with the drawback that the number of logical qubits is considerably lower than the physical ones. The process of computing which physical qubits need to be combined to form one logical qubit is called finding a *minor embedding*. If a coupling matrix Q has a zero diagonal, then we can consider it to be the adjacency matrix of a weighted graph, which we also refer to as the problem graph. When solving multiple problems with the quantum annealer, the minor embedding of the problem graph onto the hardware graph is usually recomputed before the annealing takes place. For the physical realization of the minor embedding, one has to add terms to the Hamiltonian that enforce similar behavior within the so-called *chain* of physical qubits that are combined into a logical qubit. The *chain strength* describes how strongly unequal measurement values (in the computational basis) for qubits inside the chain are penalized. The default way to calculate the minor embedding is described by [52].

In this work, we introduce a hybrid method that does not make use of minor embeddings but instead optimizes only over a subset of edges of the problem graph that are directly present in the hardware graph. In other words, we use the hardware graph as a mask to pick edges of the problem graph for the quadratic terms of the subproblems and linearize the rest. In order to take care of all parts of the problem graph, one regularly changes the correspondence between the problem and the hardware graph via a random permutation. To build upon the computations of the last iteration before the permutation of the hardware graph, one introduces a damping term. While we did not find an easy recipe to predict for which cases our iterative method should be preferred over minor embedding methods, it is obvious that our method can be applied to larger problems and has less difficulty with highly connected graphs. Methods that generate subproblems by keeping some variables fixed and optimizing over the rest are an important comparison for our iterative algorithm. Whether these methods perform better, irrespective of possible minor embedding problems, is very dependent on the size of the subproblems. We also want to stress that the method presented here, as well as many subproblem methods [219], has the drawback that there is no guarantee of getting global optimal solutions even if the underlying hardware-restricted QUBO solver were perfect. However, we will also sketch a rough idea of how it might be possible to extend these methods for global optimization by adding stochastic terms that make the resulting method non-monotone.

In the following section, we will refer to some publications where computing minor embeddings plays a crucial role or where one iteratively solves subproblems. Then we introduce the new method in Sec. 8.5 and present some theoretical results about its weak monotonicity in Sec. 8.5.1. Finally, in Sec. 8.6, we present experiments to extensively benchmark the presented method against a large neighborhood local search of various sub-problem sizes. We also show and analyze problems that cannot be addressed using minor embeddings (i.e. of comparably large sizes). Next, we present the results of proof-of-concept implementations of our method on the actual D-Wave hardware.

8.4 Related Work

Many articles in the literature aim at improving the calculation of minor embeddings or consider special cases of problem graphs. On the other hand, numerous works exist in the mathematical optimization and operation research communities on how to solve large optimization problems in an iterative, piece-wise, and often heuristic manner. We will discuss now how some of these

works are connected to the optimization with D-Wave quantum annealers.

Hardware embeddings The problem of finding a minor embedding naturally arises for hardware-restricted quantum annealing solvers. For the D-Wave machines, this led to the necessity to research fast and practical algorithms to calculate minor embeddings. In 2014, Cai et al. published a first heuristic for this problem, while most previous research dealt with exact algorithms with calculable time complexity [52]. Subsequent work often makes it possible to compute embeddings with fewer physical qubits even faster [113, 214]. Designing such heuristics is crucial for the success of quantum annealing via minor embeddings, as the task of finding an embedding for the typical hardware graphs in D-Wave machines is NP-hard [167]. This statement depends on the fact that, usually, there are defective qubits present which have to be accounted for. On the other hand, the question of whether a certain graph is embeddable can be answered in polynomial time [222] if the hardware graph is fixed. Another option is to precompute minor embeddings, e.g. for cliques [43] or for cubic lattices [145]. However, if the actual problem to solve in the end has fewer edges than, e.g. the fully connected problem, one typically has worse performance compared to specially tailored embeddings.

Decomposition, preprocessing or iterative methods Besides minor embeddings, divide-and-conquer [118] or decomposition approaches, e.g. [216], are common ways to deal with large QUBOs. More specifically, Ref. [118] views the couplings of the QUBO as a graph that is divided into subgraphs such that the weights of the edges within the subgraph are strong compared to the weights of possible edges that go out of the subgraph - an approach called *community detection*. Then, the resulting inter- and extra-community interactions are treated separately.

If one encounters Ising problems that are initially too large for the quantum annealer, one can also try to reduce the number of qubits with preprocessing techniques like in Ref. [253]. The idea is to identify sets of vertices that will have the same value assigned to them in the final solution. By treating these sets as single vertices, one can set up a smaller, reduced problem. In Ref. [253], the preprocessing technique was applied to Max-Cut problems from the MQLib dataset [95].

Due to their equivalence to QUBOs, Max-Cut problems are often considered for benchmarking in quantum optimization. For example, in Ref. [96] there is a particularly exhaustive comparison of different methods, including a Quantum Approximate Optimization Algorithm (QAOA) method on a gate-based superconducting quantum computer. Furthermore, it is also possible to compute several problems on the quantum annealer in parallel [215]. Another publication where multiple decomposition methods are investigated is [27]. One insight that was gathered there is that it is often preferable not to recompute the minor embedding for every small subproblem in iterative algorithms.

The work most related to ours is Ref. [219]. Here, the authors use the quantum annealer with a large-neighborhood local-search hybrid algorithm. They mostly consider lattice-like problems in which the embeddings are trivial or can be precomputed easily. We, on the other hand, focus more on datasets with diverse connectivity patterns or even fully connected problems: Our proposed method works independently of the connectivity structure of the problem instance.

Classical optimization and constraints Our method falls in the same category as many proximal gradient descent methods that were developed for composite optimization problems that

occur in compressive sensing or image denoising. In particular, one obstruction to using simple gradient descent algorithms for these applications is that the objective often has a part that is not differentiable. An excellent monograph to learn about proximal algorithms can be found in Parikh et al. [210]. From this source, we will introduce the reader to several facts about this topic. First, the definition of a *proximal operator* is the following: it maps between real-valued functions as $f \mapsto \text{prox}_f$ with

$$\text{prox}_f(v) = \underset{x \in \mathbb{R}^n}{\arg\min} \left(f(x) + \frac{1}{2}||x - v||_2^2 \right). \tag{8.3}$$

This operator can be viewed as a generalization of a projection on a set. As already mentioned, one is interested in composite optimization problems of the form

$$\min_{x \in \mathbb{R}^n} f(x) + g(x), \tag{8.4}$$

where f and g are closed, proper convex functions. $f : \mathbb{R}^n \to \mathbb{R}$ is assumed to be differentiable, while $g : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ does not have to be differentiable and is often thought of as a way to incorporate constraints via a characteristic function of a set. The proximal gradient descent algorithm for this, formulated as in Ref. [210], is given by

$$x^{(k+1)} = \text{prox}_{\lambda^{(k)} g} \left(x^{(k)} - \lambda^{(k)} \nabla f(x^{(k)}) \right) \tag{8.5}$$

with an iteration dependent step size parameter $\lambda^{(k)}$. Inserting the formula for the proximal operator yields

$$\begin{aligned} x^{(k+1)} &= \underset{x \in \mathbb{R}^n}{\arg\min} \left(\lambda^{(k)} g(x) + \frac{1}{2}||x - (x^{(k)} - \lambda^{(k)} \nabla f(x^{(k)}))||_2^2 \right) && (8.6) \\ &= \underset{x \in \mathbb{R}^n}{\arg\min} \Bigg(\lambda^{(k)} g(x) + \lambda^{(k)} \langle \nabla f(x^{(k)}), x - x^{(k)} \rangle + \frac{1}{2}||x - x^{(k)}||_2^2 \\ &\quad + \frac{1}{2}||\lambda^{(k)} \nabla f(x^{(k)})||_2^2 \Bigg), && (8.7) \end{aligned}$$

where the last summand is irrelevant for the arg min. In the context of compressive sensing, the function $g(x)$ is often $||x||_1$ to encourage sparsity of a solution vector. Since the proximal operator of the 1-norm reduces the absolute value of the individual components, popular algorithms that tackle this problem with proximal gradient descent are called the Iterative Shrinkage Threshold Algorithm (ISTA) or Fast Iterative Shrinkage Threshold Algorithm (FISTA) (s. [30] and the references therein). An extension to FISTA for the non-convex case is studied in Ref. [201]. We are not aware of extensions of proximal gradient descent to discrete optimization that are relevant here.

The Alternating Direction Method of Multipliers (ADMM) is also interesting in our context. Here, one also looks at composite problems, but f and g can be non-differentiable and can incorporate constraints. This method is useful if $f + g$ is hard to optimize directly, but the individual parts are easy to optimize.

The publication [108] investigates how to use QUBO solvers for mixed binary optimization problems and applies ADMM for this purpose. In Ref. [269], ADMM is also used for integer

programming and binary quadratic programming as a special case. One main idea here is that the binary vector constraints are written as the intersection of a hypersphere and a hypercube.

Although this is related to our use of proximal gradient descent algorithm, they aim to extend QUBO-solvers to mixed binary optimization solvers and are not concerned with hardware restrictions in quantum annealers. Another method concerned with iteratively solving constrained optimization problems on the D-Wave quantum annealer is developed in Ref. [274]. If one considers a concrete class of problems to be solved with the D-Wave machines, one is well advised to spend some time on considering efficient ways to formulate the problem as a QUBO problem [105, 237, 184]. In the easiest case, this can be done through eliminating variables by solving for them in one of the equality constraints and inserting the solution.

Furthermore, there are even more relevant classical methods to solve QUBO problems. A relaxation-based method for solving binary quadratic unconstrained problems with some similarities to our method can be seen in Ref. [202]. In Merz and Freisleben [187], a greedy algorithm as well as 1-Opt and the generalization k-Opt are evaluated on QUBO instances. The k-Opt algorithm goes in each iteration to the binary vector that differs from the current binary vector by no more than k entries and has the best energy among these. We will compare against k-Opt algorithms in the experiments section.

8.5 Methods

In this section, we describe the proposed iterative optimization method. Our method does not need a minor embedding and is motivated by a weak monotonicity proof that guarantees convergence (s. Sec. 8.5.1). Additionally, we investigate how multiple known methods are related to ours. In particular, projected gradient descent (s. Sec. 8.5.4) and local search in the sense of optimizing over subproblems while fixing the other variables (s. Sec. 8.5.3) are relevant. In Sec. 8.5.8, we present a regularization for optimizing over subproblems with a concept similar to combining simulated annealing and local search as in Ref. [176].

8.5.1 Linearization and Damping Term (Splitting)

The main idea of our method is to split the coupling matrix A from the Ising problem (8.2) into two parts, $A = A_{\text{quad}} + A_{\text{lin}}$, keep the original (quadratic) costs for A_{quad}, and linearize the costs involving A_{lin}.

Therefore, we want to optimize over quadratic functions of the form

$$E(s) = s^T A s = s^T A_{\text{quad}} s + s^T A_{\text{lin}} s, \tag{8.8}$$

where A, A_{quad}, and A_{lin} are symmetric matrices in $\mathbb{R}^{n\times n}$. The matrix A_{quad} could be equal to A in all the entries that correspond to a hardware graph from some quantum annealer architecture. A bias term $b^T s$ can, of course, also be present, but we omit it here for better readability. We propose to use the proximal gradient descent algorithm (8.7) for minimizing the quadratic function in equation (8.8), where we choose $f(x) = s^T A_{\text{lin}} s$ and $g(x) = s^T A_{\text{quad}} s + \Xi_{\{-1,1\}}(x)$, where $\Xi_{\{-1,1\}}(x)$ is the indicator function of the nonconvex set that restricts every entry of x to be in

$\{-1, 1\}$. Consequently, we obtain

$$\begin{aligned} s^{(k+1)} &= \underset{s\in\{-1,1\}^n}{\arg\min} \quad s^T A_{\text{quad}} s + (s^{(k)})^T A_{\text{lin}} s^{(k)} + \langle 2A_{\text{lin}} s^{(k)}, s - s^{(k)}\rangle + \frac{\lambda}{2}\|s - s^{(k)}\|_2^2 \\ &= \underset{s\in\{-1,1\}^n}{\arg\min} \quad s^T A_{\text{quad}} s + \langle 2A_{\text{lin}} s^{(k)}, s\rangle - \lambda\langle s, s^{(k)}\rangle. \end{aligned} \tag{8.9}$$

The above scheme is based on the fact that

$$\frac{\partial}{\partial s_i}(s^T A s) = (2As)_i \tag{8.10}$$

holds for any symmetric matrix A.

The regularization term $\lambda\,||s - s^{(k-1)}||$ from the iteration step (8.9) can be interpreted as a damping or regularization term to penalize strong deviations from the last iterate. In the following, we present a weak monotonicity proof for a setting which is more general than in our equation (8.9). A proof like this can, for example, be found in Ref. [31].

Weak monotonicity proof

We consider a composite objective function of the form

$$E(x) = E_1(x) + E_2(x)\,, \tag{8.11}$$

which we want to optimize over a set $C \subseteq \mathbb{R}^n$. For the individual parts, we assume that E_1 can be optimized directly while E_2 cannot, but we have access to its gradient ∇E_2. To do this, we take the parts where the quadratic function exists and can calculate the gradient there analytically. To solve the minimization problem

$$\min_{x\in C} E(x) \quad \text{for } E(x) = E_1(x) + E_2(x), \tag{8.12}$$

we linearize the E_2 part and solve the minimization iteratively. In more detail, we iterate in the same way as in equation (8.7):

$$\begin{aligned} x^{(k+1)} = \underset{x\in C}{\arg\min} \quad & E^{(k)}(x) \coloneqq E_1(x) + E_2(x^{(k)}) \\ & + \langle \nabla E_2(x^{(k)}), x - x^{(k)}\rangle + \frac{\lambda}{2}\|x - x^{(k)}\|^2. \end{aligned} \tag{8.13}$$

Obviously, $E^{(k)}(x^{(k)}) = E(x^{(k)})$. Moreover, for any x and sufficiently smooth E we can apply the Taylor expansion of $-E_2$ with remainder estimation to obtain

$$\begin{aligned} E^{(k)}(x) - E(x) &= -E_2(x) + E_2(x^{(k)}) + \langle \nabla E_2(x^{(k)}), x - x^{(k)}\rangle + \frac{\lambda}{2}||x - x^{(k)}||^2 \\ &= \big\langle \frac{1}{2}\nabla^2 E_2(\xi)(x - x^{(k)}), x - x^{(k)}\big\rangle + \frac{\lambda}{2}||x - x^{(k)}||^2, \end{aligned} \tag{8.14}$$

with a suitable $\xi \in C$. In case $||\nabla^2 E_2(\xi)|| \leq c$ holds for a constant c for all $\xi \in C$, we know that

$-\frac{1}{2}\langle\nabla^2 E_2(\xi)(x-x^{(k)}), x-x^{(k)}\rangle \geq -\frac{c}{2}||x-x^{(k)}||^2$. Thus, we conclude that for a regularization parameter $\lambda \geq c$

$$E^{(k)}(x) \geq E(x) \quad \forall x \in C\,. \tag{8.15}$$

This bound implies that

$$E(x^{(k+1)}) \leq E^{(k)}(x^{(k+1)}) \leq E^{(k)}(x^{(k)}) = E(x^{(k)}), \tag{8.16}$$

and we have proven monotonicity.

8.5.2 Statement for Ising Problems

We will now discuss some results that are specific to the case with discrete, binary variables. Note again that we have $E_2(s) = s^T A_{\text{lin}} s$ in this case. For the regularization term we know that weak monotonicity will be guaranteed for a regularization parameter λ greater or equal than the largest eigenvalue of A_{lin}, since $||\nabla^2 E_2(\xi)|| = ||2A_{\text{lin}}||$, assuming A_{lin} is symmetric. The exact approximation that has been made after equation (8.14) can be described in the following way:

$$\begin{aligned} &s^T A_{\text{lin}} s - (s^{(k)})^T A_{\text{lin}} s^{(k)} - 2s^{(k)} A_{\text{lin}}(s - s^{(k)}) \\ &= s^T A_{\text{lin}} s + (s^{(k)})^T A_{\text{lin}} s^{(k)} - 2s^{(k)} A_{\text{lin}} s \\ &= (s - s^{(k)})^T A_{\text{lin}}(s - s^{(k)}) \end{aligned} \tag{8.17}$$

and the absolute value of the term was upper bounded as $(s - s^{(k)})^T A_{\text{lin}}(s - s^{(k)}) \leq \|A_{\text{lin}}\| \left\|s - s^{(k)}\right\|^2$ to obtain an estimate for the regularization parameter λ.

One insight from equation (8.17) is that if A_{lin} has zeros on the diagonal, then for steps with exactly one changed entry, the approximation is perfect.

Furthermore, since we are interested in the discrete case, it is also comparatively easy to provide a convergence analysis. Although this clearly does not hold for every interesting case, we assume here that every binary vector has a different energy, so that $E(s_1) = s_1^T A s_1 + b^T s_1 \neq s_2^T A s_2 + b^T s_2 = E(s_2)$ for $s_1 \neq s_2$. From the previous analysis, it is clear that the sequence of energies $\left(E(s^{(k)})\right)_{k\in\mathbb{N}}$ is weakly monotonically decreasing. Since there are only finitely many possibilities for how the energy can decrease and one cannot go back to higher energies, the sequence of energies will converge. By excluding the case that two binary vectors have the same energy, the sequence of iterates also converges.

8.5.3 Fixing Variables, Large Neighborhood Local Search (LNLS)

A special case of the described method consists in fixing some variables and optimizing over the rest, as it is done in Raymond et al. [219] or in Refs. [29, 225] to solve large problems. In each iteration k, we pick an index set $\mathcal{J}_k \subseteq \{1, 2, \ldots, n\}$ describing the variables we want to optimize. The other variables have per definition indices in $\mathcal{J}_k^c$ and are fixed to their previous value. Therefore, we consider the optimization problem

$$s^{(k)} = \underset{\{s\in\{-1,1\}^n | \forall j\in\mathcal{J}_k^c:\ s_j = s_j^{(k-1)}\}}{\arg\min} s^T A s \tag{8.18}$$

to get the next iterate. Explicitly writing down the sums yields

$$s^{(k)} = \underset{\{s\in\{-1,1\}^n | s|_{\mathcal{J}_k^c} = s^{(k-1)}|_{\mathcal{J}_k^c}\}}{\arg\min} \sum_{i\in\mathcal{J}_k}\sum_{j\in\mathcal{J}_k} (s_i)^T A_{i,j} s_j + \sum_{i\in\mathcal{J}_k}\sum_{j\in\mathcal{J}_k^c} \left((s_i^{(k-1)})^T A_{i,j} s_j + (s_j)^T A_{j,i} s_i^{(k-1)} \right) + \text{constant}, \tag{8.19}$$

where $s|_{\mathcal{I}}$ indicates that one only looks at the vector elements for indices in the indexset $\mathcal{I}$. For symmetric A this reduces to

$$s^{(k)} = \underset{\{s\in\{-1,1\}^n | s|_{\mathcal{J}_k^c} = s^{(k-1)}|_{\mathcal{J}_k^c}\}}{\arg\min} \sum_{i\in\mathcal{J}_k}\sum_{j\in\mathcal{J}_k} (s_i)^T A_{i,j} s_j + \sum_{i\in\mathcal{J}_k}\sum_{j\in\mathcal{J}_k^c} 2(s_i^{(k-1)})^T A_{i,j} s_j. \tag{8.20}$$

Note that this method is weakly monotonic, since we always optimize the original energy and just restrict the set of binary vectors that can be obtained for the next iterate. The fact that we do not need damping for weak monotonicity here can be understood by looking at equation (8.17). If we apply the absolute value on both sides of the equation, we see how much the linearization of $s^T A_{\text{lin}} s$ differs from the original term. Here, A_{lin} has a block with zeros (or can be permuted in such a way), and outside of this block variables do not change, so that $s_j = s_j^{(k)}$. As a result the term on one side of the equation equals 0.

Fixing a set of variables and optimizing over the remaining m variables can also be called large neighborhood local search with m-flip neighborhood [112]. The m-flip neighborhood is the set of binary vectors, where not more than m entries change with respect to the current iterate. This is why we will abbreviate this method to 'LNLS' for large neighborhood local search, where m is given in the numerical experiments.

8.5.4 Relation to Projected Gradient Descent

Let us consider the special case with $A_{\text{quad}} = 0$ and $A_{\text{lin}} = A$. Now the iteration step becomes:

$$\begin{aligned} s^{(k+1)} &= \underset{s\in\{-1,1\}^n}{\arg\min} \langle 2As^{(k)}, s\rangle - \lambda\langle s, s^{(k)}\rangle \\ &= \underset{s\in\{-1,1\}^n}{\arg\min} -\langle s, s^{(k)} - 2\frac{1}{\lambda} As^{(k)}\rangle. \end{aligned} \tag{8.21}$$

The arg min together with the scalar product with $-s$ can be seen as taking the sign of every component of $\lambda s^{(k)} - 2As^{(k)}$. Furthermore, one can describe this as a projection onto the vertices of the hypercube $\{-1,1\}^n$. The term that is projected is the last iterate minus the gradient of the function we want to optimize times a constant. Therefore, we have found a relation to the projected gradient descent formula. However, it has to be noted that we do not operate on a convex set, but instead on individual points. This restriction makes it difficult to apply various results about the projected gradient descent algorithm to our case. The relation from ISTA to projected gradient descent holds even in the general case, as can be seen in Ref. [30].

Multiple relaxations leading to different gradients.

Notice that the diagonal entries of the matrix A in equation (8.2) do not matter for the optimization problem if we optimize over -1 and 1 as entries. However, for the linearized term $\langle 2A_{\text{lin}}s^k, s\rangle$ they have an effect in places where $s_i^{(k)} \neq s_i$. If the last two terms of equation (8.9) were separate, this would lead to projected gradient descent. We will now further investigate the effect of the diagonal elements. The objective function

$$f(s) = s^T As + b^T s \tag{8.22}$$

with variables $s_i \in \{-1, 1\}$ can have multiple possible relaxations to variables $s_i \in [-1, 1]$. We can see this by adding and subtracting a sum of arbitrary constants c_i:

$$\begin{aligned} f(s) &= s^T As + b^T s + \sum_i c_i - \sum_i c_i \\ &= s^T As + b^T s + \sum_i c_i s_i^2 - \sum_i c_i. \end{aligned} \tag{8.23}$$

Inserting the s_i^2 as factor does not lead to changes for $s_i \in \{-1, 1\}$ but yields a relaxation with different values in $s_i \in (-1, 1)$. Considering a relaxation also allows us to look at gradients. For the gradient we have

$$\nabla f(s) = 2s^T A + b + 2c \odot s \tag{8.24}$$

with $c = (c_1, ..., c_n)$ and "$\odot$" denoting element-wise multiplication. Since we want to treat every variable in the same way, we set $c_1 = c_2 = ... = c_n$. However, we can imagine that there is some better way to make use of the c_i parameters.

8.5.5 Permutating the Adjacency Matrix of the Hardware Graph

Instead of always linearizing the same parts of A, we can permute the adjacency matrix M of the hardware graph in each iteration k with a random permutation $P^{(k)}$. The principle is visualized in Fig. 8.1. Applying this permutation is important because the linearization is an approximation, and varying the subproblems we approximate leads to better results. Therefore, we extend the iteration step from Eq. (8.9) to:

$$A_{\text{quad}}^{(k)} := ((P^{(k)})^T M P^{(k)}) \odot A \tag{8.25}$$

$$A_{\text{lin}}^{(k)} := A - A_{\text{quad}}^{(k)} \tag{8.26}$$

$$s^{(k+1)} = \underset{s\in\{-1,1\}^n}{\arg\min}\ s^T A_{\text{quad}}^{(k)} s + \langle 2A_{\text{lin}}^{(k)} s^{(k)}, s\rangle - \lambda\langle s, s^{(k)}\rangle. \tag{8.27}$$

The whole process can be found in Algorithm 5. The regularization parameter λ can be chosen as described in Sec. 8.5.1. However, in experiments, we often encounter situations where the method is stagnant. To improve results, we allow for λ parameters smaller than $||A_{\text{lin}}||$ from Sec. 8.5.1. The next section goes into detail about how to choose the regularization parameter λ.

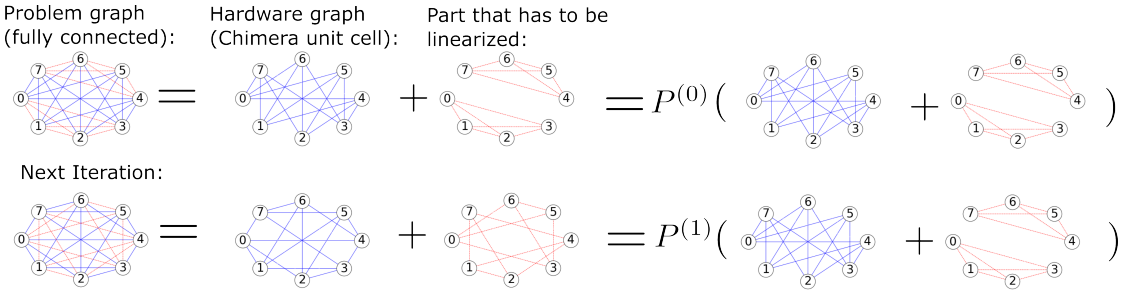


Figure 8.1: The first row shows the decomposition of a fully connected problem into a part that fits the hardware graph of a Chimera graph unit cell and a part of the edges that do not fit. In the second row, a random permutation is applied to the hardware graph to get a different decomposition. The edges without resemblance in the hardware graph influence only the biases in our method. Because of the random permutation, the part that needs to be approximated changes in every iteration.

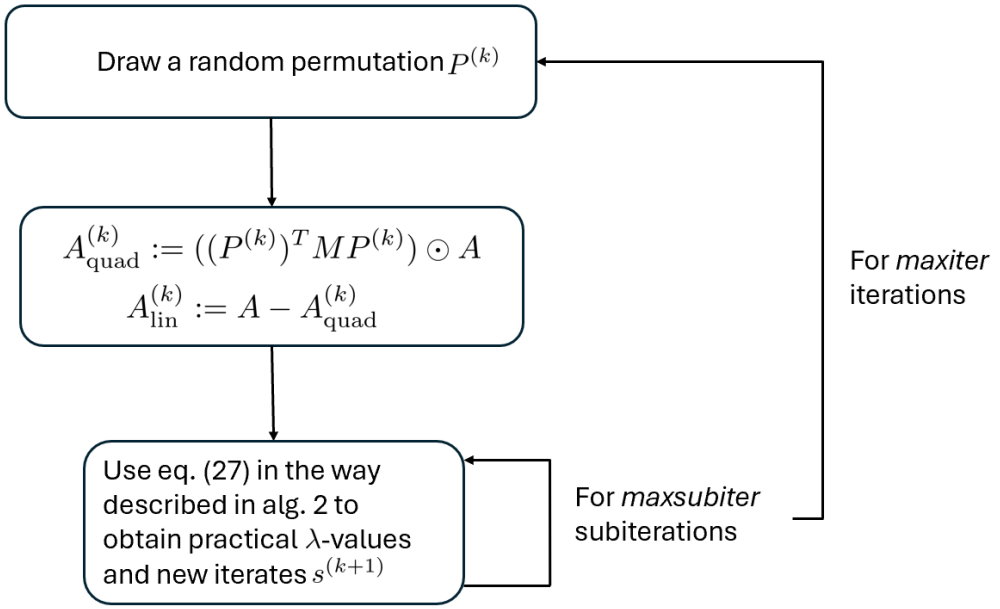


Figure 8.2: Workflow diagram of our method. In the iterations, we change the permutations, while the λ parameters are dealt with in the subiterations.

Algorithm 5 Drawing a new permutation every time

Require: Adjacency matrix of hardware graph $M \in \{0,1\}^{n\times n}$ and coupling matrix $A \in \mathbb{R}^{n\times n}$, Solver: $S(B,c) = \arg\min_{x\in\{-1,1\}^n} x^T Bx + c^T x$ for $B \in \mathbb{R}^{n\times n}$ with $B_{i,j} = 0$ if $M_{i,j} = 0$
Ensure: Try to minimize $x^T Ax$
1: Randomly initialize x_A
2: **for** $i \leftarrow 0$ **to** maxiter **do**
3: $x_\mathrm{Aold} = x_A$
4: Sample a random permutation P_k
5: $A_\mathrm{quad} = M \odot (P_k^T A P_k)$
6: $A_\mathrm{lin} = P_k^T A P_k - A_\mathrm{quad}$
7: $x_\mathrm{A} = P_k S(A_\mathrm{quad}, P_k^T((P_k A_\mathrm{lin} P_k^T)\, x_\mathrm{Aold} - \lambda x_\mathrm{Aold}))$
8: **end for**

8.5.6 Strategy for Regularization Parameters

To get practical values for the regularization parameter λ, we have a closer look at the linear term only:

$$L := P_k^T((P_k A_\mathrm{lin} P_k^T)\, x_\mathrm{Aold} - \lambda x_\mathrm{Aold}. \tag{8.28}$$

In the special case $A = A_\mathrm{lin}$, the sign of the entries in this vector determines if the iterate has changed entries. The idea for determining λ values is to order the entries of $|L|$ by size and denote this as $|L|_\mathrm{ord}$. Then, the values we try out for lambda are restricted to:

$$\lambda \in S_\lambda \subseteq S_\text{sign-change} := \left\{ v \in \mathbb{R} \mid \exists i \in \{0,...,n\} : v = \frac{(|L|_\mathrm{ord})_i + (|L|_\mathrm{ord})_{i+1}}{2} \right\}. \tag{8.29}$$

The goal of this construction is to avoid trying out multiple values for the regularization parameter λ that result in the same next iterate. The process for regularization parameter selection can be seen in Algorithm 6. The algorithm is meant to be executed in Algorithm 5 as replacement for line 7 and line 3 is removed. In Fig. 8.2 there is a workflow diagram of the method, showing the nested for-loops.

8.5.7 Choosing a Part of the Hardware Graph with high Connectivity

If the problem matrix A is smaller than the full hardware graph, we have to consider a subset of the qubits. This is done considering only qubits that belong to a Pegasus graph with a smaller size parameter n [42]. In Fig. 8.3, the adjacency matrices for Pegasus graphs with different sizes and what random permutations of this look like can be seen.

8.5.8 Simulated Annealing-like Regularization

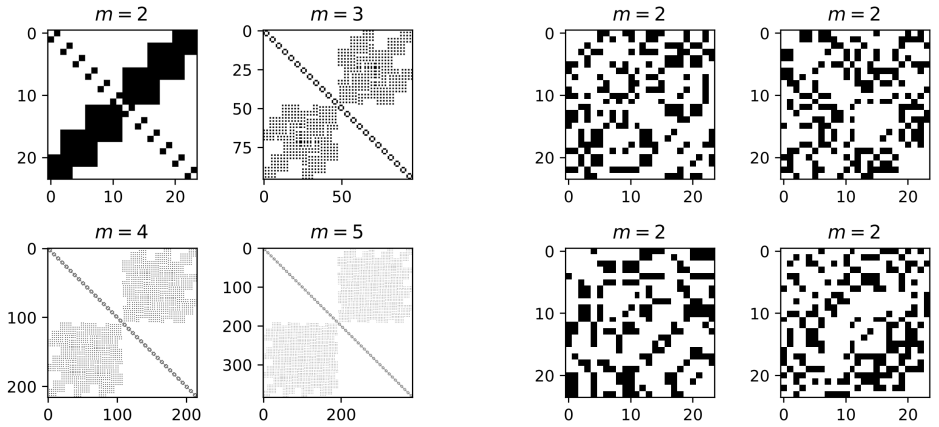
While monotonicity is a very important property for the theoretical investigation of an optimization algorithm, it has the drawback that one can get stuck in local minima. How often or if this occurs is, of course, dependent on the neighborhood structure and the problem instances themselves. Therefore, in many settings, it is often beneficial to allow steps where the energy does not decrease to escape local optima with some probability. Simulated annealing, in particular, is designed so

Algorithm 6 How to deal with λ parameters.

Require: Adjacency matrix of hardware graph $M \in \{0,1\}^{n\times n}$, coupling matrix $A \in \mathbb{R}^{n\times n}$ and current permutation P

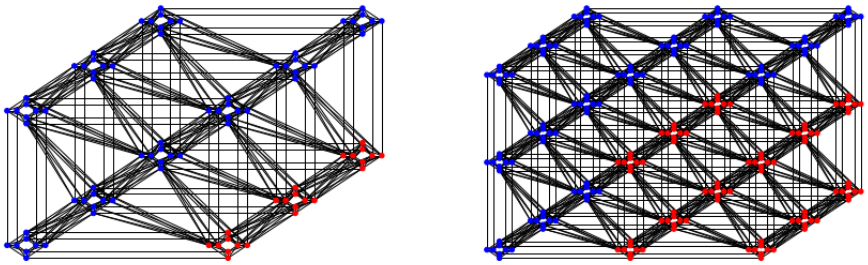
Ensure: Determine good λ parameters

Compute $S_{\text{sign-change}}$ as in (8.29)
S_λ contains every $|S_{\text{sign-change}}|$/maxsubiter-th element from $S_{\text{sign-change}}$ ordered according to size
energyBest $= E(x_{\text{Aold}})$
$x_{best} = x_{\text{Aold}}$
for $i \leftarrow 0$ **to** maxsubiter **do**
    $\lambda = (S_\lambda)_i$
    $x_{\text{Anew}} = P_k S(A_{\text{quad}}, P_k^T((P_k A_{\text{lin}} P_k^T)\, x_{\text{Aold}} - \lambda x_{\text{Aold}}))$
    if energyBest $\geq E(x_{\text{Anew}})$ **then**
        $x_{best} = x_{\text{Anew}}$
        energyBest $= E(x_{\text{Anew}})$
    end if
end for
$x_{\text{Aold}} = x_{best}$



(a) (b)

Figure 8.3: (a) Adjacency matrices for different sizes of the Pegasus graph. (b) Adjacency matrices for a Pegasus graph of size 2 but randomly permuted.



(a) (b)

Figure 8.4: The smaller Pegasus sub-graphs indicated by the red vertices in (a) and (b), respectively, are part of the bigger Pegasus graphs that also include blue vertices.

that it allows steps that do not improve the energy. In the following, we show that it is possible to formulate simulated annealing as a series of simple binary, linear optimization problems. We will use the same idea to break the monotonicity of optimizing random subproblems in an attempt to make it more likely to get to the global optimum. Let $s^{(k-1)}$ be the current iterate. A random neighbor of $s^{(k-1)}$ is constructed by first drawing a random index set $\mathcal{J}_k$ and then flipping all values that correspond to indices in $\mathcal{J}_k$. This vector we name $s_N^{(k)}$. Now the next iterate is constructed with

$$s^{(k)} = \underset{\{s\in\{-1,1\}^n | s|_{\mathcal{J}_k^c} = s^{(k-1)}|_{\mathcal{J}_k^c}\}}{\arg\min} \sum_{i\in\mathcal{J}_k} (e^{-\frac{E(s_N^{(k)}) - E(s^{(k-1)})}{T(k)}} - R_k) s_i s_i^{(k-1)}, \tag{8.30}$$

where $R_k \in [0,1]$ is for each iteration uniformly random generated and $T(k)$ is a typical temperature schedule for simulated annealing. An example for such a schedule would be $T(k) = \frac{c}{\log(1+k)}$ with a fixed constant $c \in \mathbb{R}$. The idea is to combine this with our QUBO solver like this:

$$s^{(k)} = \underset{\{s\in\{-1,1\}^n | s|_{\mathcal{J}_k^c} = s^{(k-1)}|_{\mathcal{J}_k^c}\}}{\arg\min} s^T A s + \sum_{i\in\mathcal{J}_k} (e^{-\frac{E(s_N^{(k)}) - E(s^{(k-1)})}{T(k)}} - R_k) s_i s_i^{(k-1)}. \tag{8.31}$$

One can even add factors to both summands to weight the contributions of the simulated annealing and the non-linear QUBO term. In principle, this should yield an optimization algorithm similar to the combination of simulated annealing and local search proposed in Ref. [176]. The simulated annealing part of the algorithm can only jump to vectors that have optimal energy for a fixed Hamming distance to the previous iterate. We leave the numerical verification of this idea to future research since there is still some thought required in finding the best weighting factors and a good schedule. However, we still present it here since it fits the discussion of damping terms in these iterative algorithms.

Abbreviation	Name	Reference
Splitting	Here presented method based on proximal gradient descent	s. Secs. 8.5.1, 8.5.5 and 8.5.6
LNLS	Large Neighborhood Local Search	s. Sec. 8.5.3
SA on full problem	Application of the simulated annealing solver Neal	[75]
SA hardware-restricted	Neal solver artificially restricted to D-Wave Pegasus topology	[75]
k-Opt	k-Opt with k-swap neighborhood	[187]

Table 8.1: Abbreviations and references for all the tested methods.

8.6 Results

In the following, we demonstrate the performance of our method by using simulated annealing as a QUBO sampler and artificially restricting it to the D-Wave Pegasus Hardware Graph. For the experiments with simulated annealing, we used the D-Wave neal solver [75] with 100 runs per problem. We also demonstrate the functionality of our method on the D-Wave Advantage machine version 6.4. All the methods that were tested and the section or reference where they were described can be found in Tab. 8.1.

8.6.1 Datasets

As problem instances, we investigate the regular spinglass instances from [183] and part of the Max-cut problems from MQLib [95], which we formulate as QUBOs.

In Ref. [183] among other QUBOs, a fully-connected regular spin-glass model is introduced for benchmarking. The publication also provides comparisons between the hybrid-quantum method qbsolv [4], a direct solution with the D-Wave Advantage using minor embeddings, and a baseline exhaustive search. They also experiment with Ising problems with randomly drawn coefficients. This is similar to the well-known Sherrington-Kirkpatrick model [240] but with uniform probability distributions instead of a Gaussian distribution.

The regular spin-glass model, Ising problems have the form

$$A_{i,j} = 1 - \frac{i+j-2}{N-1}, \text{ for } i \neq j \tag{8.32}$$

$$b_i = 1 - 2\frac{i-1}{N-1}, \tag{8.33}$$

for a fixed dimension N. We abbreviate these instances with "Reg".

In a Max-Cut problem, one seeks to find a cut through a weighted graph so that the sum of the weights of the edges that are cut is as large as possible. Max-Cut optimization problems are fully equivalent to QUBOs. However, for the formulation of an arbitrary QUBO problem with n variables, a Max-Cut problem with $n+1$ vertices is needed [94]. The problem instances in the MQlib dataset range from applications in very large-scale integrated circuit design and

telecommunication networks, to integrative biological network analysis or image segmentation [95]. Multiple instances are also generated using graph theoretical, mathematical models, for example, via Watz–Strogatz graphs.

8.6.2 Regular Spinglas Dataset from Ref. [183]

We will benchmark the methods presented here on the regular spinglas problems from Mehta et al. [183]. For the subproblems, a simulated annealing solver is used; for our splitting method, the connectivity is restricted to the Pegasus Hardware graph.

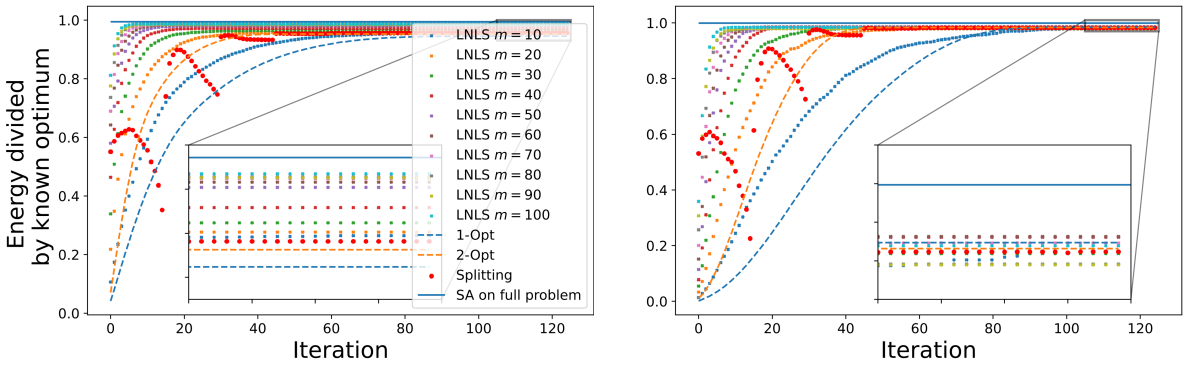
Figure 8.5 depicts how the averaged approximation ratio $\frac{E}{E_{\text{opt}}}$ decreases over the iterations. One iteration is one call of the simulated annealing QUBO solver Neal [75]. Therefore, the subiterations from Algorithm 6 are also visible in the plot. For all the reported experiments, we have used 15 as the maximal number of subiterations.

For the calculation of the approximation ratio, the ground state needs to be known. Here we use a conjecture from Ref. [183] about the ground state, which we prove in appendix Sec. 8.8.1. The conjecture states that the optimal vector for the particular instances has all the negative entries at the beginning, and after the negative entries, there are only positive entries. Knowing this, one can easily compute the exact ground state by trying out all $n + 1$ possibilities. The term splitting in the legend of 8.5 refers to our new proposed method. 1-Opt and 2-Opt refer to iteratively optimizing brute force in a 1 or 2 swap neighborhood, as we already mentioned in the related work Sec. 8.4.

The experiments show that our method can handle bigger instances, which do not fit on the hardware graph completely, much better. Restricting the number of instances to those that are bigger than 150, as is done in Fig. 8.5 (b), improves the performance of the splitting compared to plainly solving subproblems. The same statement can be made for the k-Opt methods. Note that the smallest size of the subproblems is 10, which is already the size of the smallest instances. We also used a simulated annealing solver that is artificially restricted to the hardware connectivity and obtained solutions via minor embedding. This is named "SA hardware-restricted" in the plots. In Fig. 8.6 (a), we plot the behavior for all instances where an embedding is found. Note that although the Reg instances are fully connected, we did not use pre-computed embeddings but the default minorminer code. The time consumption of calculating minor embeddings can easily exceed the time the other methods take. This is depicted in the (b) part of Fig. 8.6.

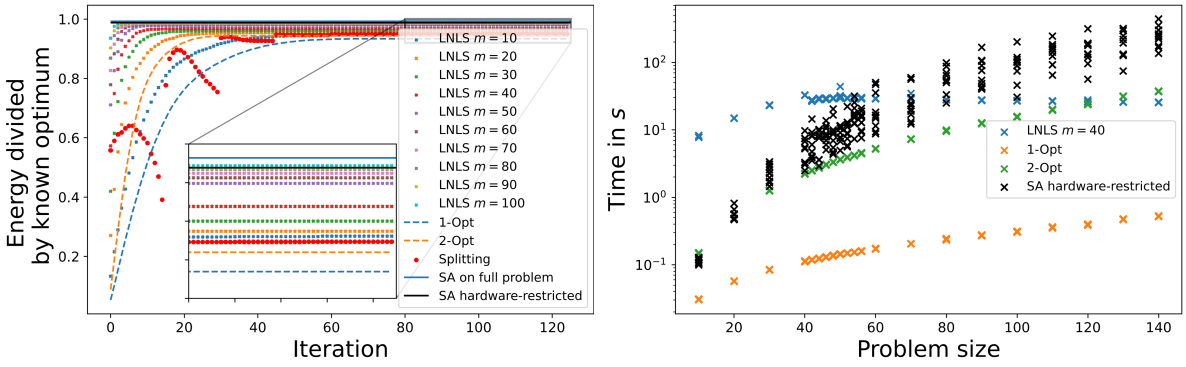
8.6.3 Maximum Cut Problems

We test our methods on some of the instances from MQLib [95]. While the MQLib contains in total 3396 instances, we only look at 909 instances with problem sizes between 500 and 1000. The maximum cut problem is equivalent to a QUBO, as can be seen, for example, in Ref. [94]. We compare the performance on these instances for varying dimensionality m of the LNLS (s. Sec. 8.6.3). In Sec. 8.6.3 we also show results from runs on the D-Wave Advantage quantum annealer. For the other experiments, we always used simulated annealing solvers. In the appendix Sec. 8.8.2, we run an ablation study on this dataset to compare with a method that randomly applies permutations to the hardware graph without taking previous iterates into account and then outputs the best result.



(a) (b)

Figure 8.5: Approximation ratio depending on the iteration, e.g. number of neal QUBO-solver calls, averaged over (a) all Reg instances, (b) all Reg instances greater than 150. The plot shows how the optimization progresses for the different methods.



(a) (b)

Figure 8.6: (a) Approximation ratio for all instances where an embedding could be found. An embedding was found in 210 from the 270 Reg instances. The time limit for finding an embedding was 10min. (b) Wall clock time consumption for the different methods on the Reg instances. The method where the minor embedding is computed can be found as “SA hardware-restricted" in the legend. Although the “SA hardware-restricted” method outputs almost the best energies, it also has the worst time consumption for the bigger instances.

Comparison with various subproblem sizes

We tested our splitting method as well as the plain calculation of subproblems with different sizes on the 922 problems from MQlib that have sizes between 500 and 1000 vertices. Results of this can be seen in Fig. 8.7. In the approximation factor plot, we divide the absolute energy that is obtained by the heuristic with the best reported energy in Ref. [95]. The 13 instances where the best reported energy is 0 are not used for this plot. In the last iteration, the splitting method performed on par with optimization over subproblems of size 30. Calculating bigger subproblems achieved better energy, while using smaller subproblems achieved worse energies.

Note that the problems are already so big that one cannot find a minor embedding to the D-Wave Advantage machine. The 2-Opt optimization problem also does not finish in a reasonable amount of time for the big instances. That computation time is a bottleneck here for 2-Opt is to be expected since the time scales with $\mathcal{O}(\frac{n^2}{2})$. Nevertheless, we also computed the 2-Opt method for completeness, and the results can be seen in the appendix Sec. 8.8.3 in Fig. 8.12.

Furthermore, we present the final energy of each instance divided by the best heuristic for each instance in Figs. 8.8 and 8.9. For this depiction, the instances were ordered according to the performance of our splitting method.
It can be observed that a good performance compared to the best heuristics is strongly correlated among the different methods. Further analysis revealed that the instances where LNLS or our splitting method performed the best are often given by denser graphs. It also appears to be beneficial to have edge weights that are restricted to attaining only a few discrete values. Among the instances where the here presented splitting method coincides with the best heuristic were multiple Barabasi Albert [25] and Powerlaw cluster [127] random graphs. On one really sparse and two very dense instances, our method performed better than the best heuristics that we found in the MQLib repository [95]. But since for these three instances, even LNLS with $m = 10$ achieved better results, we do not want to focus on this point. The instances are named “g000098", “g002324" and “g002064". For the instances where our method performed worst compared to other heuristics, we observed a trend that those instances have rather sparse graphs. We assume that in these cases, other heuristics can better make use of the specific, sparse structure. A lot of the difficult instances are described to be models for solving real-world Very-Large-Scale Integrated Circuit [150] problem. The instance where our method performed worst was a hypercube graph [220, 39].

Implementation on D-Wave Advantage

We tested our methods on the D-Wave Advantage machine with version 6.4. Throughout the experiments, we used the default schedule and the default annealing time of $20\mu s$. The chain strength is chosen to be of factor 1.001 stronger than the biggest absolute value in the couplings or biases. From the obtained state histogram, we only work with the state of lowest energy that has been obtained. Note that while even stronger chain strengths can reduce the chain break frequency for some problems, they also lead to a reduced effective precision, which can impair performance. If chain breaks occur, they are dealt with by using the default majority voting strategy. Overall, we have observed that in our benchmark, a chain strength of 1.001 is beneficial over significantly larger values.

By trying out different instances, we came to the conclusion that it is important what degree of precision is required for the QUBO matrix. This is well known, and methods to make a QUBO

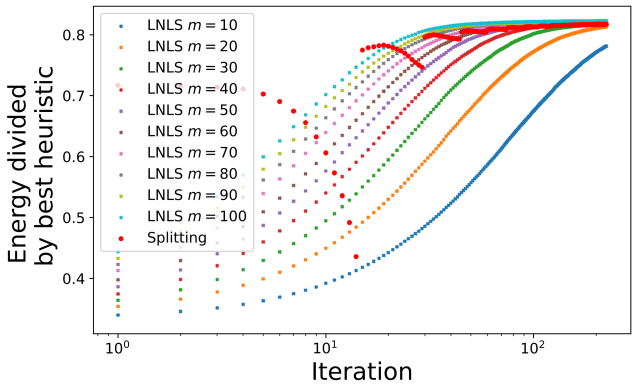
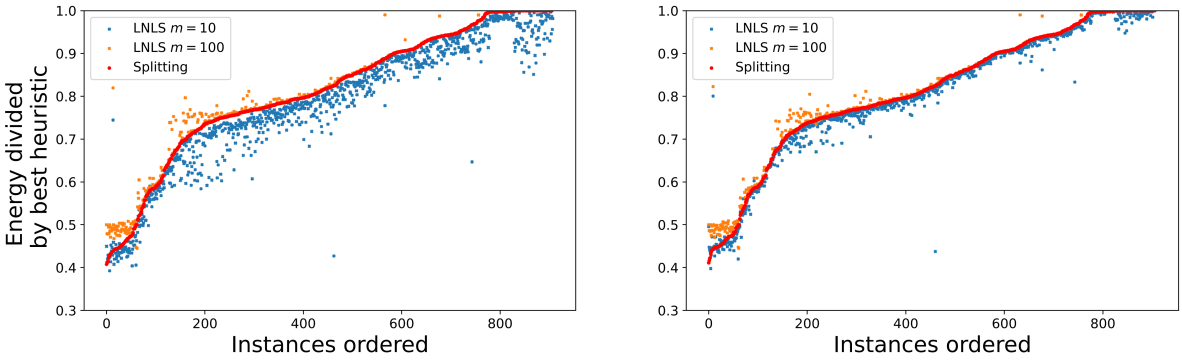
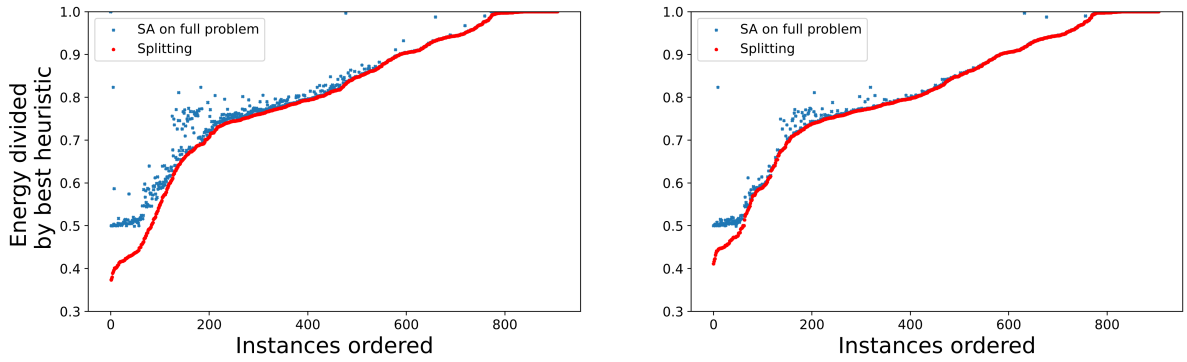


Figure 8.7: Approximation factor averaged over 909 Instances from MQlib depending on the iteration. Which method performs best largely depends on the size of the subproblems. Bigger subproblems lead to faster improvement in the energy on average.



(a) (b)

Figure 8.8: Results for all the instances compared to the best heuristic for (a) 15 and (b) 25 iterations. The instances are ordered according to the performance of our splitting method. It can be observed that the performance compared to the best heuristic is largely correlated among the methods.



(a) (b)

Figure 8.9: Results for all the instances compared to the best heuristic for (a) 3 and (b) 25 iterations of the splitting method. The instances are ordered according to the performance of our splitting method. The performance of simulated annealing without hardware restrictions is also correlated to the performance of our splitting method compared to the best heuristic. Note that the values presented for the unrestricted simulated annealing are the same on both plots but can be reordered according to the performance of the splitting method.

problem better suited for precision-restricted solvers like the D-Wave machines with experimental errors in the couplings have already been developed [190]. We observed failure cases where our method never improved the energy or even made it worse. For example, it failed to optimize the instance "imgseg_271031", which is motivated by an image segmentation problem and has float-valued weights in the graph. But also, the instance "gka.5f" with integer values between -50 and 50 was not optimized properly. At this point, to circumvent the problem, we tried out several instances with $-1, 1$ as possible values for the edges and could confirm that our method works on the quantum annealer. The energy decreased in the same way as we observed in the simulated annealing experiments. In Fig. 8.10, the average over 13 instances of this sort is depicted.

8.7 Conclusion

The splitting techniques from our method were, for the first time, investigated for discrete QUBO optimization. We obtained the theoretical insight that this method can be constructed weakly monotonically. Nevertheless, for the actual experiments, the method was more practical in a non-monotonically decreasing setting to avoid stagnant behavior. The comparison with the standard large neighborhood local search in this setting showed that the better performance is mostly dependent on the size of the local neighborhood. Most of our benchmark instances were too large for a direct minor embedding of the problem, but it is well known that these methods can take a long time and can lead to large chains that decrease performance. This was also confirmed for the part of the regular spin glass instances that was small enough. Whether calculating minor embeddings or using iterative algorithms is the better strategy is, of course, highly dependent on the problem, and we cannot give a final answer to this. The question of which characteristics of the problem instances can predict whether the method with splitting

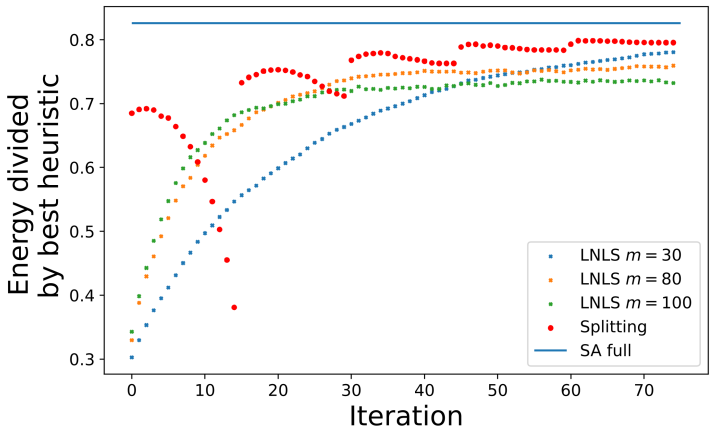


Figure 8.10: Energies divided by the best reported heuristic averaged for 13 Instances of MQlib that have $-1, 1$ as coefficients. The experiments were carried out on the D-Wave Advantage quantum annealer except for "SA full". Therefore, we have provided a proof of concept for our iterative splitting method.

performs better than the other iterative methods is also largely unanswered. However, in the experiments using the actual quantum device, we obtained promising results on some Max-Cut problems that have $+1$ and -1 as coefficients. In general, we hope this work encourages further research in splitting methods for QUBO optimization via hardware-restricted solvers. We will publish our source code at the repository [6].

8.8 Appendix

8.8.1 Proof for the Conjecture about Regular Spin Glasses from Ref. [183]

The problem instances are constructed according to

$$A_{i,j} = 1 - \frac{i+j-2}{N-1}, \text{ for } i \neq j \tag{8.34}$$

$$b_i = 1 - 2\frac{i-1}{N-1}. \tag{8.35}$$

In the following, we prove their conjecture about the groundstate, which states that the groundstates always have a form where the -1 entries occur first, and after the last -1, there are just $+1$ entries. First, note that their coupling matrices can be written as:

$$A = \mathbb{E} - C - R, \tag{8.36}$$

where

$$\mathbb{E}_{i,j} = 1 \tag{8.37}$$

$$C_{i,j} = \frac{i-1}{N-1} \tag{8.38}$$

$$R_{i,j} = \frac{j-1}{N-1}. \tag{8.39}$$

Now consider the following

$$\begin{aligned}
&\arg\min_{s\in\{-1,1\}} s^T(\mathbb{E} - C - R)s + b^T s && (8.40)\\
&= \arg\min_{s\in\{-1,1\}} s^T\mathbb{E}s - s^T Cs - s^T Rs + b^T s && (8.41)\\
&= \arg\min_{s\in\{-1,1\}} s^T\mathbb{E}s - s^T C^T s - s^T Rs + b^T s && (8.42)\\
&= \arg\min_{s\in\{-1,1\}} s^T(\mathbb{E} - 2R)s + b^T s. && (8.43)
\end{aligned}$$

Written out, one has to optimize

$$\begin{aligned}
&\sum_i (1 - 2\frac{i-1}{N-1})s_i + \sum_{i,j}\sum_{i\neq j} (1 - \frac{i+j-2}{N-1})s_i s_j && (8.44)\\
&= \sum_i (1 - 2\frac{i-1}{N-1})s_i + \sum_{i,j}\sum_{i\neq j} (1 - \frac{2i-2}{N-1})s_i s_j && (8.45)\\
&= \sum_i s_i(1 + \sum_{j\neq i} s_j)(1 - \frac{2(i-1)}{N-1}). && (8.46)
\end{aligned}$$

Now we add the term $\sum_i s_i^2(1 - \frac{2(i-1)}{N-1})$, which is constant in s_i, since $s_i^2 = 1$. This yields

$$(1 + \sum_j s_j)\sum_i s_i(1 - \frac{2(i-1)}{N-1}) \tag{8.47}$$

which we want to optimize. Note first that the first factor only depends on the number of variables that are +1 and not on which ones have the value. The second factor is maximized or minimized for

$$s_{+-} := (1, 1, , ..., 1, -1, ..., -1) \tag{8.48}$$

or

$$s_{-+} := (-1, -1, , ..., -1, 1, ..., 1) = -s_{+-}. \tag{8.49}$$

Additionally we know that

$$s^T As = (-s)^T A(-s). \tag{8.50}$$

This tells us that the question of whether s or $-s$ has a better energy only depends on the linear term. We will show that $b^T s_{+-} \geq 0$. This would then yield that $b^T s_{+-} \leq 0$ which concludes the proof that the minimizer is always of the form s_{-+}. Let k denote the index where the signs change in the vector, then

$$\begin{aligned}
b^T s_{+-} &= \sum_{i=1}^{k}(1 - 2\frac{i-1}{N-1}) - \sum_{i=k+1}^{N}(1 - 2\frac{i-1}{N-1}) \\
&= k - 2\frac{1}{N-1}(-k + \sum_{i=1}^{k} i) - (N - k - 2\frac{1}{N-1}(-(N-k) + \sum_{i=k+1}^{N} i)) \\
&= k - 2\frac{1}{N-1}(-k + \frac{k(k+1)}{2}) - (N - k - 2\frac{1}{N-1}(-(N-k) \\
&+ (\frac{N(N+1)}{2} - \frac{k(k+1)}{2}))) \\
&= 2(k - 2\frac{1}{N-1}(-k + \frac{k(k+1)}{2})) - (N - 2\frac{1}{N-1}(-N + (\frac{N(N+1)}{2}))) \\
&= 2(k - 2\frac{1}{N-1}(\frac{k(k-1)}{2})) - (N - 2\frac{1}{N-1}(\frac{N(N-1)}{2})) \\
&= 2(k - \frac{k(k-1)}{N-1}) - (N - N) \\
&= 2k(1 - \frac{k-1}{N-1}) \geq 0 \quad \forall k \in \{1, ..., N\}.
\end{aligned} \tag{8.51}$$

Therefore, we know that vectors s_{+-} have always worse or equal energy than vectors s_{-+}, and the complete statement from [183] is proven. The problem instances which are publicly accessible from Ref. [183] only are upper triangular, which is also indicated in their publication. But the proof works in the same way as for the convention with symmetric matrices by just introducing a factor $\frac{1}{2}$ in equation (8.34).

8.8.2 Ablation Study

We will show an experiment to confirm that the damping term in equation (8.9) improves the result. For this, we compare against the same method with the regularization parameter λ set to zero. For the comparison method, where λ is set to zero, we draw a new permutation in every subiteration. This is the case since in the original algorithm, we would only optimize the λ parameter within the subiterations. The experiment on 922 Maximum cut problems can be seen in Fig. 8.11. For the $\lambda = 0$ case, the lowest energy from all the subiterations was used.

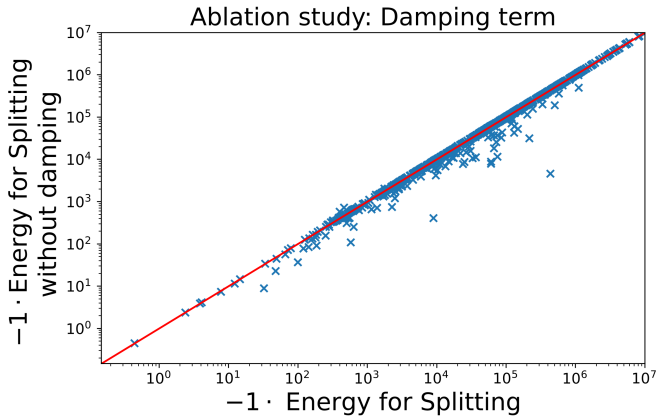


Figure 8.11: Comparison of the final objective function values (energies) with the method without damping, i.e. $\lambda = 0$. From the 922 problem instances with sizes between 500 and 1000 vertices, in 156 cases both methods achieved the same result. In 383 cases, the version with no damping was better and in 383 cases, the version with damping was better. However, the method with damping is even better than the method without in 296 cases if you multiply it by a factor 1.001. The reverse statement is only true in 164 cases. This implies that for instances where no damping had a better result, the improvement is rather small.

8.8.3 MQlib Experiments with k-Opt

For the 909 MQlib instances that we investigated in Sec. 8.6.3, we also computed for 1-Opt and 2-Opt the average approximation ratios depending on the iteration number. This can be seen in Fig. 8.12.

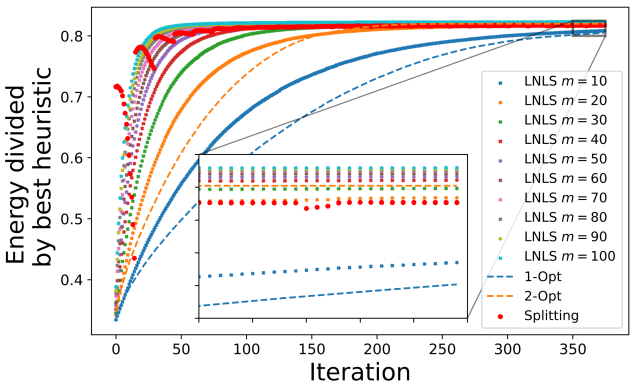


Figure 8.12: Approximation ratio of 1-Opt and 2-Opt for 909 MQLib instances included in Fig. 8.7.

Acknowledgment

We thank Natacha Kuete Meli for careful proofreading of the manuscript and helpful discussion. MK is funded by Fujitsu Germany GmbH as part of the endowed professorship “Quantum Inspired and Quantum Optimization”.

Chapter 9

Conclusion and Discussion

This thesis dealt with the possibility of using quantum annealing for combinatorial optimization problems in computer vision. It is of course hard to make a definite conclusion if there ever will come a time that quantum annealers will be used in this way. From a theoretical perspective it is hard to get runtime estimates, since e.g. the spectral gap is difficult to compute. Our experiments on the current quantum annealing hardware highlighted several challenges that need to be overcome. One is the limited size and connectivity of the quantum annealers and another challenge are the experimental errors in the couplings. The thesis presented a couple of algorithmic considerations that will likely be even interesting for arbitrary QUBO solvers. Since in most applications constraints are present, the use of unrestricted solvers requires that one spends some effort in setting up the penalty parameters and terms. Furthermore, iterative local search like methods like those presented in the Q-Match or the CCuantuMM publications often outperform direct QUBO formulations if there is a strong size restriction for the underlying solver. Finally using neural networks to set up the most beneficial formulation of a combinatorial optimization problem is also a promising avenue of research and in general hybrid algorithms that incorporate quantum annealers only for some subproblems are very relevant in the forseeable future. All in all, we are excited for further developments and progress in the field of quantum annealing and quantum computing. If in some experiments it is reported that quantum computing outperforms classical computing, one has to carefully analyze and design the classical competing algorithms. Thus quantum computing research and classical algorithm design are mutually beneficial to each other. At the moment of writing this thesis the free availability of some quantum annealing computing time has already led in multiple fields to publications, where first cornerstones are layed down to explore the applicability of quantum annealing. We believe that this curiosity driven research is fruitful no matter what the final developments of this technology will be and are waiting eagerly to learn about future breakthroughs.

Bibliography

[1] IBM Q experience. https://quantum-computing.ibm.com/ http://www.research.ibm.com/quantum. Accessed: 2020 January and February.

[2] Ice: Dynamic ranges in h and j values. https://docs.dwavesys.com/docs/latest/c_qpu_1.html. Accessed: 2020 July.

[3] Performance tuning for d-wave quantum processors. https://www.dwavesys.com/sites/default/files/2_Wed_Am_PerfTips.pdf. Accessed: 2020 July.

[4] qbsolv - d-wave ocean software documentation. https://docs.ocean.dwavesys.com/projects/qbsolv/en/latest/intro.html. [Accessed 11-07-2024].

[5] QPU Solvers: Minor-Embedding; D-Wave System Documentation documentation — docs.dwavesys.com. https://docs.dwavesys.com/docs/latest/handbook_embedding.html. [Accessed 20-08-2024].

[6] Source code. https://github.com/MSeelbach/SplittingForQA. [Accessed 11-07-2024].

[7] Source code for dwave.system.composites.tiling. https://docs.ocean.dwavesys.com/projects/system/en/stable/_modules/dwave/system/composites/tiling.html. Accessed: 2022-02-09.

[8] S. H. Adachi and M. P. Henderson. Application of quantum annealing to training of deep neural networks. *arXiv preprint arXiv:1510.06356*, 2015.

[9] Y. Aflalo, A. Bronstein, and R. Kimmel. On convex relaxation of graph isomorphism. *Proceedings of the National Academy of Sciences*, 112(10):2942–2947, 2015.

[10] D. Aharonov, W. Van Dam, J. Kempe, Z. Landau, S. Lloyd, and O. Regev. Adiabatic quantum computation is equivalent to standard quantum computation. *SIAM review*, 50(4):755–787, 2008.

[11] T. Albash and D. A. Lidar. Adiabatic quantum computation. *Reviews of Modern Physics*, 90(1):015002, 2018.

[12] M. H. S. Amin. Consistency of the adiabatic theorem. *Phys. Rev. Lett.*, 102:220401, Jun 2009.

[13] E. Andriyash and M. H. Amin. Can quantum monte carlo simulate quantum annealing? *arXiv preprint arXiv:1703.09277*, 2017.

[14] D. Anguelov et al. The correlated correspondence algorithm for unsupervised registration of nonrigid surfaces. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2004.

[15] K. M. Anstreicher and N. W. Brixius. Solving quadratic assignment problems using convex quadratic programming relaxations. *Optimization Methods and Software*, 16(1-4):49–68, 2001.

[16] J.-P. Antoine, R. Bishop, A. Bohm, and S. Wickramasekara. Rigged hilbert spaces in quantum physics. 07 2009.

[17] F. Arrigoni and A. Fusiello. Synchronization problems in computer vision with closed-form solutions. *Int. J. Comput. Vis.*, 128:26–52, 2020.

[18] F. Arrigoni, W. Menapace, M. S. Benkner, E. Ricci, and V. Golyanik. Quantum motion segmentation. In *Eur. Conf. Comput. Vis.*, 2022.

[19] F. Arrigoni and T. Pajdla. Motion segmentation via synchronization. In *IEEE International Conference on Computer Vision Workshops (ICCVW)*, 2019.

[20] F. Arrigoni and T. Pajdla. Robust motion segmentation from pairwise matches. In *Int. Conf. Comput. Vis.*, 2019.

[21] F. Arute et al. Quantum supremacy using a programmable superconducting processor. *Nature*, 2019.

[22] M. Aubry, U. Schlickewei, and D. Cremers. The wave kernel signature: A quantum mechanical approach to shape analysis. In *Int. Conf. Comput. Vis. Worksh.*, 2011.

[23] C. Aytekin, S. Kiranyaz, and M. Gabbouj. Automatic object segmentation by quantum cuts. In *International Conference on Pattern Recognition*, 2014.

[24] F. Bach. Submodular functions: from discrete to continuous domains. *Mathematical Programming*, 175(1):419–459, 2019.

[25] A.-L. Barabási and R. Albert. Emergence of scaling in random networks. *science*, 286(5439), 1999.

[26] D. Barath and J. Matas. Multi-class model fitting by energy minimization and mode-seeking. In *Eur. Conf. Comput. Vis.*, pages 229–245. Springer International Publishing, 2018.

[27] G. Bass, M. Henderson, J. Heath, and J. Dulny III. Optimizing the optimizer: decomposition techniques for quantum annealing. *Quantum Machine Intelligence*, 3(1), 2021.

[28] J. Batle, C. Ooi, A. Farouk, M. Abutalib, and S. Abdalla. Do multipartite correlations speed up adiabatic quantum computation or quantum annealing? *Quantum Information Processing*, 15(8):3081–3099, 2016.

[29] J. E. Beasley. Heuristic algorithms for the unconstrained binary quadratic programming problem, 1998.

[30] A. Beck and M. Teboulle. Fast gradient-based algorithms for constrained total variation image denoising and deblurring problems. *IEEE transactions on image processing*, 18(11), 2009.

[31] A. Beck and M. Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM journal on imaging sciences*, 2(1), 2009.

[32] K. Beer, D. Bondarenko, T. Farrelly, T. J. Osborne, R. Salzmann, D. Scheiermann, and R. Wolf. Training deep quantum neural networks. *Nature Communications*, 11(808), 2020.

[33] F. Bernard, C. Theobalt, and M. Moeller. Ds*: Tighter lifting-free convex relaxations for quadratic matching problems. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2018.

[34] F. Bernard, J. Thunberg, P. Swoboda, and C. Theobalt. Hippi: Higher-order projected power iterations for scalable multi-matching. In *Int. Conf. Comput. Vis.*, 2019.

[35] E. Bernstein and U. Vazirani. Quantum complexity theory. In *Annual ACM Symposium on Theory of Computing*, 1993.

[36] D. P. Bertsekas. A distributed algorithm for the assignment problem. Laboratory for Information and Decision Systems Working Paper, M.I.T., 1979.

[37] D. P. Bertsekas. Network Optimization: Continuous and Discrete Models. Athena Scientific, 1998.

[38] J. Biamonte, P. Wittek, N. Pancotti, P. Rebentrost, N. Wiebe, and S. Lloyd. Quantum machine learning. *Nature*, 549, 11 2016.

[39] I. Biazzo, A. Braunstein, and R. Zecchina. Performance of a cavity-method-based algorithm for the prize-collecting steiner tree problem on graphs. *Physical Review E—Statistical, Nonlinear, and Soft Matter Physics*, 86(2), 2012.

[40] T. Birdal, V. Golyanik, C. Theobalt, and L. Guibas. Quantum permutation synchronization. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2021.

[41] F. Bogo, J. Romero, M. Loper, and M. J. Black. FAUST: Dataset and evaluation for 3D mesh registration. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2014.

[42] K. Boothby, P. Bunyk, J. Raymond, and A. Roy. Next-Generation Topology of D-Wave Quantum Processors. *arXiv e-prints*, 2020.

[43] T. Boothby, A. D. King, and A. Roy. Fast clique minor generation in chimera qubit connectivity graphs. *Quantum Information Processing*, 15, 2016.

[44] M. Born and V. Fock. Beweis des adiabatensatzes. *Zeitschrift für Physik*, 51(3):165–180, 1928.

[45] E. Boyda, S. Basu, S. Ganguly, A. Michaelis, S. Mukhopadhyay, and R. R. Nemani. Deploying a quantum annealing processor to detect tree cover in aerial imagery of california. *PLoS ONE*, 12, 2017.

[46] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 23(11), 2001.

[47] A. M. Bronstein, M. M. Bronstein, and R. Kimmel. *Numerical geometry of non-rigid shapes*. Springer Science & Business Media, 2008.

[48] M. M. Bronstein and I. Kokkinos. Scale-invariant heat kernel signatures for non-rigid shape recognition. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2010.

[49] O. Burghard and R. Klein. Efficient lifted relaxations of the quadratic assignment problem. In *Proceedings of the Conference on Vision, Modeling and Visualization*, 2017.

[50] R. E. Burkard, S. E. Karisch, and F. Rendl. Qaplib - a quadratic assignment problem library. *Journal of Global Optimization*, 1997.

[51] R. E. Burkard and J. Offermann. Entwurf von schreibmaschinentastaturen mittels quadratischer zuordnungsprobleme. *Zeitschrift für Operations Research*, 1977.

[52] J. Cai, W. G. Macready, and A. Roy. A practical heuristic for finding graph minors. *arXiv e-prints*, 2014.

[53] C. Cao, J. Xue, N. Shannon, and R. Joynt. Speedup of the quantum adiabatic algorithm using delocalization catalysis. *arXiv preprint arXiv:2007.11212*, 2020.

[54] D. Cao and F. Bernard. Unsupervised deep multi-shape matching. In *Eur. Conf. Comput. Vis.*, 2022.

[55] S. Caraiman and V. I. Manta. Image processing using quantum computing. In *International Conference on System Theory, Control and Computing (ICSTCC)*, 2012.

[56] A. Carlier, K. Leonard, S. Hahmann, G. Morin, and M. Collins. The 2d shape structure dataset: A user annotated open access database. *Computers & Graphics*, 58:23–30, 2016.

[57] D. Castelvecchi. Quantum computers ready to leap out of the lab in 2017. *Nature News*, 541(7635):9, 2017.

[58] G. Cavallaro, D. Willsch, M. Willsch, K. Michielsen, and M. Riedel. Approaching remote sensing image classification with ensembles of support vector machines on the d-wave quantum annealer. In *IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, 2020.

[59] Q. Chen and V. Koltun. Robust nonrigid registration by convex optimization. In *International Conference on Computer Vision*, 2015.

[60] A. M. Childs, E. Farhi, and J. Preskill. Robustness of adiabatic quantum computation. *Physical Review A*, 65(1):012322, 2001.

[61] T.-J. Chin, D. Suter, S.-F. Ch'ng, and J. Quach. Quantum robust fitting. In *Proceedings of the Asian Conference on Computer Vision (ACCV)*, November 2020.

[62] M. Cho, K. Alahari, and J. Ponce. Learning graphs to match. In *International Conference on Computer Vision*, pages 25–32, 2013.

[63] J. Choo. Investigating the feasibility of solving the quadratic assignment problem using quantum computing. `https://joelchoo.com/research/qc-paper.pdf`. Accessed: 2020 July.

[64] P. J. Coles et al. Quantum algorithm implementations for beginners. *arXiv preprint arXiv:1804.03719*, 2018.

[65] R. Computing. https://www.rigetti.com/, 2022.

[66] L. Cosmo, E. Rodola, A. Albarelli, F. Mémoli, and D. Cremers. Consistent partial matching of shape collections via sparse modeling. In *Computer Graphics Forum*, volume 36, pages 209–221. Wiley Online Library, 2017.

[67] E. Crosson and A. W. Harrow. Simulated quantum annealing can be exponentially faster than classical simulated annealing. In *2016 IEEE 57th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 714–723. IEEE, 2016.

[68] D-Wave. Operation and timing. https://docs.dwavesys.com/docs/latest/c_qpu_timing.html, 2021.

[69] D-Wave. Zephyr topology of d-wave quantum processors (d-wave technical report series). https://www.dwavesys.com/media/2uznec4s/14-1056a-a_zephyr_topology_of_d-wave_quantum_processors.pdf, 2021.

[70] D-Wave Systems. Technical Description of the D-Wave Quantum Processing Unit. https://docs.dwavesys.com/docs/latest/doc_qpu.html, 2019. online; accessed 5.9.2019 and 3.2.2020.

[71] D-Wave Systems. Leap. https://docs.dwavesys.com/docs/latest/leap.html, 2022. online; accessed on the 24.01.2022.

[72] I. D-Wave Systems. D-wave leap: Take the leap. https://cloud.dwavesys.com/leap/, 2020. online; accessed on the 8 April 2020.

[73] D-Wave Systems, Inc. D-wave system documentation. https://docs.dwavesys.com/docs/latest/c_gs_4.html. online; accessed on 2022-03-06.

[74] D-Wave Systems, Inc. Leap datasheet v10. https://www.dwavesys.com/sites/default/files/Leap_Datasheet_v10_0.pdf, 2021. online; accessed on the 25.02.2021.

[75] D-Wave Systems, Inc. Simulated annealing sampler. https://github.com/dwavesystems/dwave-neal/blob/master/docs/reference/sampler.rst, 2021. online; accessed on the 25.02.2021 and 6.03.2022.

[76] D-Wave Systems, Inc. Solver computation time. https://docs.dwavesys.com/docs/latest/timing_qa_cycle_time.html, 2021. online; accessed on the 15.03.2021.

[77] D-Wave Systems, Inc. D-wave ocean. https://docs.ocean.dwavesys.com/en/stable, 2022. online.

[78] D-Wave Systems, Inc. dimod simulated annealing sampler. https://test-projecttemplate-dimod.readthedocs.io/en/latest/reference/sampler_composites/samplers.html, 2022. online.

[79] D-Wave Systems, Inc. dwave-neal. `https://github.com/dwavesystems/dwave-neal`, 2022. online.

[80] D-Wave Systems, Inc. Leap, 2022.

[81] A. Das and B. K. Chakrabarti. *Quantum annealing and related optimization methods*, volume 679. Springer Science & Business Media, 2005.

[82] P. Date and T. Potok. Adiabatic quantum linear regression. *Scientific Reports*, 11, 2021.

[83] N. Dattani. Quadratization in discrete optimization and quantum mechanics. *arXiv preprint arXiv:1901.04405*, 2019.

[84] N. Dattani, S. Szalay, and N. Chancellor. Pegasus: The second connectivity graph for large-scale quantum annealing hardware. *arXiv e-prints*, 2019.

[85] B. Dema, J. ARAI, and K. HORIKAWA. Support vector machine for multiclass classification using quantum annealers. 2020.

[86] V. S. Denchev, S. Boixo, S. V. Isakov, N. Ding, R. Babbush, V. Smelyanskiy, J. Martinis, and H. Neven. What is the computational value of finite-range tunneling? *Physical Review X*, 6(3):031015, 2016.

[87] B. Deng, Y. Yao, R. M. Dyke, and J. Zhang. A survey of non-rigid 3d registration. *Computer Graphics Forum*, 41(2):559–589, 2022.

[88] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition*, pages 248–255. IEEE, 2009.

[89] T. Deprelle, T. Groueix, M. Fisher, V. G. Kim, B. C. Russell, and M. Aubry. Learning elementary structures for 3d shape generation and matching. In *Neurips*, 2019.

[90] A.-D. Doan, M. Sasdelli, D. Suter, and T.-J. Chin. A hybrid quantum-classical algorithm for robust fitting. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2022.

[91] Y. Du, M.-H. Hsieh, T. Liu, and D. Tao. Expressive power of parametrized quantum circuits. *Phys. Rev. Research*, 2:033125, 2020.

[92] V. Dunjko and H. J. Briegel. Machine learning & artificial intelligence in the quantum domain: a review of recent progress. *Reports on Progress in Physics*, 81(7):074001, 2018.

[93] V. Dunjko, J. M. Taylor, and H. J. Briegel. Quantum-enhanced machine learning. *Phys. Rev. Lett.*, 117:130501, 2016.

[94] I. Dunning, S. Gupta, and J. Silberholz. Equivalence of qubo and maxcut. `https://github.com/MQLib/MQLib/files/10934880/equivalence.pdf`. [Accessed 11-07-2024].

[95] I. Dunning, S. Gupta, and J. Silberholz. What works best when? a systematic evaluation of heuristics for max-cut and QUBO. *INFORMS Journal on Computing*, 30(3), 2018.

[96] M. Dupont, B. Sundar, B. Evert, D. E. B. Neira, Z. Peng, S. Jeffrey, and M. J. Hodson. Quantum optimization for the maximum cut problem on a superconducting quantum computer. *arXiv preprint arXiv:2404.17579*, 2024.

[97] R. Dyke, C. Stride, Y.-K. Lai, P. L. Rosin, M. Aubry, A. Boyarski, A. M. Bronstein, M. M. Bronstein, D. Cremers, and M. F. et al. Shrec'19: Shape correspondence with isometric and non-isometric deformations. In *Workshop on 3D Object Retrieval (3DOR)*, 2019.

[98] N. Dym, H. Maron, and Y. Lipman. Ds++: A flexible, scalable and provably tight relaxation for matching problems. *ACM Trans. Graph.*, 36(6), 2017.

[99] B. Eschermann and H.-J. Wunderlich. Optimized synthesis of self-testable finite state machines. In *20th International Symposium on Fault-Tolerant Computing (FFTCS 20)*, 1990.

[100] E. Farhi, J. Goldstone, and S. Gutmann. Quantum adiabatic evolution algorithms versus simulated annealing. *arXiv preprint quant-ph/0201031*, 2002.

[101] E. Farhi, J. Goldstone, S. Gutmann, J. Lapan, A. Lundgren, and D. Preda. A quantum adiabatic evolution algorithm applied to random instances of an np-complete problem. *Science*, 292(5516):472–475, 2001.

[102] A. Ferber, B. Wilder, B. Dilkina, and M. Tambe. Mipaal: Mixed integer program as a layer. *AAAI Conference on Artificial Intelligence*, 34:1504–1511, 2020.

[103] P. Flajolet and R. Sedgewick. *Analytic combinatorics*. Cambridge University Press, 2009.

[104] F. Fogel, R. Jenatton, F. Bach, and A. d' Aspremont. Convex relaxations for permutation problems. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 26, 2013.

[105] A. Fowler. *Improved QUBO formulations for D-Wave quantum computing*. PhD thesis, University of Auckland, 2017.

[106] T. Gabor, S. Feld, H. Safi, T. Phan, and C. Linnhoff-Popien. Insights on training neural networks for qubo tasks. In *Proceedings of the IEEE/ACM 42nd International Conference on Software Engineering Workshops*, pages 436–441, 2020.

[107] F. Gaitan and L. Clark. Graph isomorphism and adiabatic quantum computing. *Phys. Rev. A*, 89:022342, Feb 2014.

[108] C. Gambella and A. Simonetto. Multiblock admm heuristics for mixed-binary optimization on classical and quantum computers. *IEEE Transactions on Quantum Engineering*, 1, 2020.

[109] M. Gao, Z. Lahner, J. Thunberg, D. Cremers, and F. Bernard. Isometric multi-shape matching. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2021.

[110] S. Geman and D. Geman. Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. *IEEE Transactions on pattern analysis and machine intelligence*, (6):721–741, 1984.

[111] V. Golyanik and C. Theobalt. A quantum computational approach to correspondence problems on point sets. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2020.

[112] T. F. Gonzalez. *Handbook of approximation algorithms and metaheuristics*. Chapman and Hall/CRC, Boca Raton, Florida, 2007.

[113] T. D. Goodrich, B. D. Sullivan, and T. S. Humble. Optimizing adiabatic quantum program compilation using a graph-theoretic framework. *Quantum Information Processing*, 17(5), 2018.

[114] E. K. Grant and T. S. Humble. Adiabatic quantum computing and quantum annealing, 07 2020.

[115] T. Groueix, M. Fisher, V. G. Kim, B. Russell, and M. Aubry. 3d-coded : 3d correspondences by deep deformation. In *Eur. Conf. Comput. Vis.*, 2018.

[116] L. K. Grover. A fast quantum mechanical algorithm for database search. In *Annual ACM Symposium on Theory of Computing*, 1996.

[117] L. K. Grover. Quantum mechanics helps in searching for a needle in a haystack. *Physical review letters*, 79(2):325, 1997.

[118] G. G. Guerreschi. Solving quadratic unconstrained binary optimization with divide-and-conquer and quantum algorithms. *arXiv preprint arXiv:2101.07813*, 2021.

[119] S. W. Hadley, F. Rendl, and H. Wolkowicz. A new lower bound via projection for the quadratic assignment problem. *Mathematics of Operations Research*, 1992.

[120] P. Hahn, T. Grant, and N. Hall. A branch-and-bound algorithm for the quadratic assignment problem based on the hungarian method. *European Journal of Operational Research*, 108(3):629–640, 1998.

[121] B. Hajek. Cooling schedules for optimal annealing. *Mathematics of operations research*, 13(2):311–329, 1988.

[122] O. Halimi, O. Litany, E. R. Rodolà, A. M. Bronstein, and R. Kimmel. Unsupervised learning of dense shape correspondence. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2019.

[123] P. Hauke, H. G. Katzgraber, W. Lechner, H. Nishimori, and W. D. Oliver. Perspectives of quantum annealing: Methods and implementations. *Reports on Progress in Physics*, 83(5):054401, 2020.

[124] V. Havlíček, A. D. Córcoles, K. Temme, A. W. Harrow, A. Kandala, J. M. Chow, and J. M. Gambetta. Supervised learning with quantum-enhanced feature spaces. *Nature*, 567:209–212, 2019.

[125] M. Hebenstreit, D. Alsina, J. I. Latorre, and B. Kraus. Compressed quantum computation using a remote five-qubit quantum computer. *Phys. Rev. A*, 95:052339, May 2017.

[126] L. Hérault, R. Horaud, F. Veillon, and J.-J. Niez. Symbolic image matching by simulated annealing. In *Brit. Mach. Vis. Conf.*, 1990.

[127] P. Holme and B. J. Kim. Growing scale-free networks with tunable clustering. *Physical review E*, 65(2), 2002.

[128] B. Holzschuh, Z. Lähner, and D. Cremers. Simulated annealing for 3d shape correspondence. In *Conference of 3D Vision (3DV)*, 2020.

[129] M. Howard, J. Wallman, V. Veitch, and J. Emerson. Contextuality supplies the 'magic' for quantum computation. *Nature*, 2014.

[130] Q.-X. Huang and L. Guibas. Consistent shape maps via semidefinite programming. In *Computer graphics forum*, volume 32, pages 177–186, 2013.

[131] R. Huang, J. Ren, P. Wonka, and M. Ovsjanikov. Consistent zoomout: Efficient spectral map synchronization. In *Computer Graphics Forum*, volume 39, pages 265–278, 2020.

[132] L. Hutschenreiter, S. Haller, L. Feineis, C. Rother, D. Kainmüller, and B. Savchynskyy. Fusion moves for graph matching. In *Int. Conf. Comput. Vis.*, 2021.

[133] Ibm q experience. `https://quantum-computing.ibm.com`, 2021. Accessed on the 26.02.2021.

[134] M. Jarret and S. P. Jordan. Adiabatic optimization without local minima. *arXiv preprint arXiv:1405.7552*, 2014.

[135] B. Jiang, J. Tang, C. Ding, and B. Luo. Binary constraint preserving graph matching. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2017.

[136] N. Jiang, Y. Dang, and J. Wang. Quantum image matching. *Quantum Information Processing*, 15(9):3543–3572, 2016.

[137] J. R. Johansson, P. Nation, and F. Nori. Qutip: An open-source python framework for the dynamics of open quantum systems. *Computer Physics Communications*, 183(8):1760–1772, 2012.

[138] M. Jünger, E. Lobe, P. Mutzel, G. Reinelt, F. Rendl, G. Rinaldi, and T. Stollenwerk. Performance of a quantum annealer for ising ground state computations on chimera graphs. *arXiv preprint arXiv:1904.11965*, 2019.

[139] T. Kadowaki and H. Nishimori. Quantum annealing in the transverse ising model. *Phys. Rev. E*, 58:5355–5363, 1998.

[140] I. Kezurer, S. Z. Kovalsky, R. Basri, and Y. Lipman. Tight relaxation of quadratic matching. In *Symposium on Geometry Processing (SGP)*, 2015.

[141] K.H. Han and J.H. Kim. Genetic quantum algorithm and its application to combinatorial optimization problem. In *Congress on Evolutionary Computation*, 2000.

[142] S. Khairy, R. Shaydulin, L. Cincio, Y. Alexeev, and P. Balaprakash. Learning to optimize variational quantum circuits to solve combinatorial problems. In *AAAI Conference on Artificial Intelligence*, 2020.

[143] A. Khoshaman, W. Vinci, B. Denis, E. Andriyash, H. Sadeghi, and M. H. Amin. Quantum variational autoencoder. *Quantum Science and Technology*, 4(1), 2019.

[144] V. G. Kim, Y. Lipman, and T. Funkhouser. Blended intrinsic maps. *ACM Trans. Graph.*, 2011.

[145] A. D. King and W. Bernoudy. Performance benefits of increased qubit connectivity in quantum annealing 3-dimensional spin glasses. *arXiv preprint arXiv:2009.12479*, 2020.

[146] A. D. King and C. C. McGeoch. Algorithm engineering for a quantum annealing platform. *arXiv preprint arXiv:1410.2628*, 2014.

[147] A. D. King, J. Raymond, T. Lanting, S. V. Isakov, M. Mohseni, G. Poulin-Lamarre, S. Ejtemaee, W. Bernoudy, I. Ozfidan, A. Y. Smirnov, et al. Scaling advantage over path-integral monte carlo in quantum simulation of geometrically frustrated magnets. *Nature Communications*, 12(1):1–6, 2021.

[148] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[149] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. Optimization by simulated annealing. *SCIENCE*, 220(4598):671–680, 1983.

[150] T. Koch and A. Martin. Solving steiner tree problems in graphs to optimality. *Networks: An International Journal*, 32(3), 1998.

[151] G. Kochenberger, J.-K. Hao, F. Glover, M. Lewis, Z. Lü, H. Wang, and Y. Wang. The unconstrained binary quadratic programming problem: a survey. *Journal of combinatorial optimization*, 28(1):58–81, 2014.

[152] V. Kolmogorov. A new look at reweighted message passing. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 37(5):919–930, 2015.

[153] T. C. Koopmans and M. Beckmann. Assignment Problems and the Location of Economic Activities. *Econometrica*, 25(1):53, Jan. 1957.

[154] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 25, 2012.

[155] J. M. Kübler, S. Buchholz, and B. Schölkopf. The inductive bias of quantum kernels. In *Neural Information Processing Systems (NIPS)*, 2021.

[156] H. W. Kuhn and B. Yaw. The hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 2:83–97, 1955.

[157] C. Lanczos. An iteration method for the solution of the eigenvalue problem of linear differential and integral operators. *Journal of Research of the National Bureau of Standards*, 1950.

[158] E. L. Lawler. The quadratic assignment problem. *Management science*, 9(4):586–599, 1963.

[159] D. K. Lê-Huu and N. Paragios. Alternating direction graph matching. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2017.

[160] Y. LeCun, S. Chopra, R. Hadsell, F. J. Huang, and et al. A tutorial on energy-based learning. In *Predicting Structured Data*. MIT Press, 2006.

[161] V. Lempitsky, C. Rother, S. Roth, and A. Blake. Fusion moves for markov random field optimization. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 32(8):1392–1405, 2010.

[162] M. Leordeanu and M. Hebert. A spectral technique for correspondence problems using pairwise constraints. In *International Conference on Computer Vision (ICCV)*, 2005.

[163] J. Li and S. Ghosh. Quantum-soft qubo suppression for accurate object detection. In *Eur. Conf. Comput. Vis.*, 2020.

[164] O. Litany, T. Remez, E. Rodola, A. Bronstein, and M. Bronstein. Deep functional maps: Structured prediction for dense shape correspondence. In *International Conference on Computer Vision (ICCV)*, 2017.

[165] B. Liu, M. Wang, H. Foroosh, M. Tappen, and M. Pensky. Sparse convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 806–814, 2015.

[166] S. Liu and G. Trenkler. Hadamard, Khatri-Rao, Kronecker and other matrix products. *International Journal of Information and Systems Sciences*, 4(1):160 – 177, 2008.

[167] E. Lobe and A. Lutz. Minor embedding in broken chimera and derived graphs is np-complete. *Theoretical Computer Science*, 989, 2024.

[168] O. Lockwood and M. Si. Reinforcement learning with quantum variational circuit. *AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment (AIIDE)*, 16:245–251, 2020.

[169] S. Loncaric. A survey on shape analysis techniques. *Pattern Recognition*, 31(8), 1998.

[170] F. Lu and E. Milios. Robot pose estimation in unknown environments by matching 2d range scans. *Journal of Intelligent and Robotic systems*, 18(3):249–275, 1997.

[171] L. Magri and A. Fusiello. T-Linkage: A continuous relaxation of J-Linkage for multi-model fitting. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 3954–3961, June 2014.

[172] L. Magri and A. Fusiello. Robust multiple model fitting with preference analysis and low-rank approximation. In *Proceedings of the British Machine Vision Conference*, pages 20.1–20.12. BMVA Press, September 2015.

[173] W. Mahdi, S. A. Medjahed, and M. Ouali. Performance analysis of simulated annealing cooling schedules in the context of dense image matching. *Computación y Sistemas*, 21(3):493–501, 2017.

[174] N. Mariella and A. Simonetto. A Quantum Algorithm for the Sub-Graph Isomorphism Problem. *arXiv e-prints*, 2021.

[175] H. Maron, N. Dym, I. Kezurer, S. Kovalsky, and Y. Lipman. Point registration via efficient convex relaxation. *ACM Trans. Graph.*, 35(4), 2016.

[176] O. C. Martin and S. W. Otto. Combining simulated annealing with local search heuristics. *Annals of operations research*, 63(1), 1996.

[177] J. Masci, D. Boscaini, M. M. Bronstein, and P. Vandergheynst. Geodesic convolutional neural networks on riemannian manifolds. In *International Conference on Computer Vision Workshop (ICCVW)*, 2015.

[178] E. Maset, F. Arrigoni, and A. Fusiello. Practical and efficient multi-view matching. In *Int. Conf. Comput. Vis.*, 2017.

[179] K. Mato, R. Mengoni, D. Ottaviani, and G. Palermo. Quantum molecular unfolding. *arXiv e-prints*, 2021.

[180] J. R. McClean, S. Boixo, V. N. Smelyanskiy, R. Babbush, and H. Neven. Barren plateaus in quantum neural network training landscapes. *Nature Communications*, 9(4812), 2018.

[181] T. M. McCormick, B. R. Osborn, R. B. Angle, and R. L. Streit. Implementation of a multiple target tracking filter on an adiabatic quantum computer. In *IEEE Aerospace Conference (AERO)*, 2022.

[182] C. C. McGeoch. Adiabatic quantum computation and quantum annealing: Theory and practice. *Synthesis Lectures on Quantum Computing*, 5(2):1–93, 2014.

[183] V. Mehta, F. Jin, K. Michielsen, and H. De Raedt. On the hardness of quadratic unconstrained binary optimization problems. *Frontiers in Physics*, 2022.

[184] N. K. Meli, V. Golyanik, M. Seelbach Benkner, and M. Moeller. Qucoop: A versatile framework for solving composite and binary-parametrised problems on quantum annealers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11395–11405, June 2025.

[185] N. K. Meli, F. Mannel, and J. Lellmann. An iterative quantum approach for transformation estimation from point sets. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2022.

[186] S. Melzi, J. Ren, E. Rodolà, A. Sharma, P. Wonka, and M. Ovsjanikov. Zoomout: Spectral upsampling for efficient shape correspondence. *ACM Trans. Graph. (Proc. SIGGRAPH Asia)*, 2019.

[187] P. Merz and B. Freisleben. Greedy and local search heuristics for unconstrained binary quadratic programming. *Journal of heuristics*, 8, 2002.

[188] K. Mitarai, M. Negoro, M. Kitagawa, and K. Fujii. Quantum circuit learning. *Phys. Rev. A*, 98:032309, 2018.

[189] F. Monti, D. Boscaini, J. Masci, E. Rodola, J. Svoboda, and M. M. Bronstein. Geometric deep learning on graphs and manifolds using mixture model cnns. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2017.

[190] S. Mücke, T. Gerlach, and N. Piatkowski. Optimum-preserving qubo parameter compression, 2023.

[191] F. Neukart, G. Compostella, C. Seidel, D. von Dollen, S. Yarkoni, and B. Parney. Traffic flow optimization using a quantum annealer. *Frontiers in ICT*, 4:29, 2017.

[192] H. Neven, V. S. Denchev, G. Rose, and W. G. Macready. Training a binary classifier with the quantum adiabatic algorithm. *arXiv e-prints*, 2008.

[193] H. Neven, V. S. Denchev, G. Rose, and W. G. Macready. Qboost: Large scale classifier training with adiabatic quantum optimization. In *Asian Conference on Machine Learning (ACML)*, 2012.

[194] H. Neven, G. Rose, and W. G. Macready. Image recognition with an adiabatic quantum computer i. mapping to quadratic unconstrained binary optimization. *arXiv preprint arXiv:0804.4457*, 2008.

[195] N. T. Nguyen, G. T. Kenyon, and B. Yoon. A regression algorithm for accelerated lattice qcd that exploits sparse inference on the d-wave quantum annealer. *Scientific Reports*, 2020.

[196] N. T. T. Nguyen and G. Kenyon. Image classification using quantum inference on the d-wave 2x. In *International Conference on Rebooting Computing (ICRC)*, 2018.

[197] M. A. Nielsen and I. Chuang. Quantum computation and quantum information, 2002.

[198] M. Noormandipour and H. Wang. Matching point sets with quantum circuit learning. In *International Conference on Acoustics, Speech and Signal Processing*, 2022.

[199] C. E. Nugent, T. E. Vollmann, and J. Ruml. An experimental comparison of techniques for the assignment of facilities to locations. *Operations Research*, 1968.

[200] J. Nüßlein, C. Roch, T. Gabor, C. Linnhoff-Popien, and S. Feld. Black box optimization using qubo and the cross entropy method. *arXiv preprint arXiv:2206.12510*, 2022.

[201] P. Ochs and T. Pock. Adaptive fista for nonconvex optimization. *SIAM Journal on Optimization*, 29(4), 2019.

[202] C. Olsson, A. P. Eriksson, and F. Kahl. Improved spectral relaxation methods for binary quadratic optimization problems. *Computer Vision and Image Understanding*, 112(1), 2008.

[203] T. P. Orlando, J. E. Mooij, L. Tian, C. H. van der Wal, L. S. Levitov, S. Lloyd, and J. J. Mazo. Superconducting persistent-current qubit. *Phys. Rev. B*, 60:15398–15413, Dec 1999.

[204] R. Orus, S. Mugel, and E. Lizaso. Forecasting financial crashes with quantum computing. *Physical Review A*, 99, 06 2019.

[205] M. Ovsjanikov, M. Ben-Chen, J. Solomon, A. Butscher, and L. Guibas. Functional maps: a flexible representation of maps between shapes. *ACM Trans. Graph.*, 31(4):1–11, 2012.

[206] D. O'Malley, V. V. Vesselinov, B. S. Alexandrov, and L. B. Alexandrov. Nonnegative/binary matrix factorization with a d-wave quantum annealer. *PLOS ONE*, 13(12):1–12, 12 2018.

[207] D. Pachauri, R. Kondor, and V. Singh. Solving the multi-way matching problem by permutation synchronization. *Adv. Neural Inform. Process. Syst.*, 2013.

[208] J. Palmer. Quantum devices: Here, there and everywhere. *The Economist, Technology Quarterly, Quarter*, 2, 2017.

[209] P. M. Pardalos, F. Rendl, and H. Wolkowicz. The quadratic assignment problem: A survey and recent developments. In *DIMACS Workshop on Quadratic Assignment Problems*, volume 16. American Mathematical Society, 1994.

[210] N. Parikh, S. Boyd, et al. Proximal algorithms. *Foundations and trends® in Optimization*, 1(3), 2014.

[211] D. Pastorello and E. Blanzieri. Quantum annealing learning search for solving qubo problems. *Quantum Information Processing*, 18(10):1–17, 2019.

[212] D. Pastorello, E. Blanzieri, and V. Cavecchia. Learning adiabatic quantum algorithms over optimization problems. *Quantum Machine Intelligence*, 3(1):1–19, 2021.

[213] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019.

[214] R. Patton, C. Schuman, T. Potok, et al. Efficiently embedding qubo problems on adiabatic quantum computers. *Quantum Information Processing*, 18(4), 2019.

[215] E. Pelofske, G. Hahn, and H. N. Djidjev. Parallel quantum annealing. *Scientific Reports*, 12(1), 2022.

[216] M. Ponce, R. Herrman, P. C. Lotshaw, S. Powers, G. Siopsis, T. Humble, and J. Ostrowski. Graph decomposition techniques for solving combinatorial optimization problems with variational quantum algorithms, 2023.

[217] J. Povh and F. Rendl. Copositive and semidefinite relaxations of the quadratic assignment problem. *Discrete Optimization*, 6(3):231–241, 2009.

[218] R. Raussendorf and H. J. Briegel. A one-way quantum computer. *Phys. Rev. Lett.*, 86:5188–5191, May 2001.

[219] J. Raymond, R. Stevanovic, W. Bernoudy, K. Boothby, C. C. McGeoch, A. J. Berkley, P. Farré, J. Pasvolsky, and A. D. King. Hybrid quantum annealing for larger-than-qpu lattice-structured problems. *ACM Transactions on Quantum Computing*, 4(3), 2023.

[220] M. G. Resende, J. P. de Sousa, I. Rosseti, M. P. de Aragão, C. C. Ribeiro, E. Uchoa, and R. F. Werneck. *New benchmark instances for the Steiner problem in graphs*. Springer, New York City, New York, 2004.

[221] Rigetti Computing. Rigetti quantum cloud services. `https://docs.rigetti.com/qcs/`, 2022. online; accessed on the 24.01.2022.

[222] N. Robertson and P. Seymour. Graph minors .xiii. the disjoint paths problem. *Journal of Combinatorial Theory, Series B*, 63(1), 1995.

[223] J. Roland and N. J. Cerf. Quantum search by local adiabatic evolution. *Physical Review A*, 65(4), 2002.

[224] M. Rolinek, P. Swoboda, D. Zietlow, A. Paulus, V. Musil, and G. Martius. Deep graph matching via blackbox differentiation of combinatorial solvers. In *European Conference on Computer Vision (ECCV)*, 2020.

[225] G. Rosenberg, M. Vazifeh, B. Woods, and E. Haber. Building an iterative heuristic solver for a quantum annealer. *Computational Optimization and Applications*, 65:845–869, 2016.

[226] C. Roucairol. *Du sequentiel au parallele: la recherche arborescente et son application a la programmation quadratique en variables 0 et 1.* PhD thesis, Université Pierre et Marie Curie, 1987.

[227] C. Roucairol. A parallel branch and bound algorithm for the quadratic assignment problem. *Discrete Applied Mathematics*, 18(2):211–225, 1987.

[228] S. Rujikietgumjorn and R. T. Collins. Optimized pedestrian detection for multiple and occluded people. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2013.

[229] Y. Sahilioglu and Y. Yemez. Multiple shape correspondence by dynamic programming. *Computer Graphics Forum (CGF)*, 33(7), 2014.

[230] Y. Sahillioglu. Recent advances in shape correspondence. *The Visual Computer*, 36, 2020.

[231] S. Sahni and T. Gonzalez. P-complete approximation problems. *Journal of the ACM*, 1976.

[232] S. Salti, F. Tombari, and L. D. Stefano. Shot: Unique signatures of histograms for surface and texture description. *Comput. Vis. Image Underst. (CVIU)*, 125:251–264, 2014.

[233] M. Sasdelli and T.-J. Chin. Quantum annealing formulation for binary neural networks. *arXiv preprint arXiv:2107.02751*, 2021.

[234] C. Schellewald and C. Schnörr. Probabilistic subgraph matching based on convex relaxation. In *Energy Minimization Methods in Computer Vision and Pattern Recognition*, 2005.

[235] W. R. Scott. *Group theory.* Courier Corporation, 2012.

[236] M. Scriabin and R. C. Vergin. Comparison of computer algorithms and visual based methods for plant layout. *Management Science*, 1975.

[237] M. Seelbach Benkner, V. Golyanik, C. Theobalt, and M. Moeller. Adiabatic quantum graph matching with permutation matrix constraints. In *Conference of 3D Vision (3DV)*, 2020.

[238] M. Seelbach Benkner, Z. Lähner, V. Golyanik, C. Wunderlich, C. Theobalt, and M. Moeller. Q-match: Iterative shape matching via quantum annealing. In *Int. Conf. Comput. Vis.*, 2021.

[239] Y. Shen, Q. Huang, N. Srebro, and S. Sanghavi. Normalized spectral map synchronization. *Adv. Neural Inform. Process. Syst.*, 2016.

[240] D. Sherrington and S. Kirkpatrick. Solvable model of a spin-glass. *Physical review letters*, 35(26):1792, 1975.

[241] P. W. Shor. Algorithms for quantum computation: discrete logarithms and factoring. In *Proceedings 35th Annual Symposium on Foundations of Computer Science*, Nov 1994.

[242] S. Sim, P. D. Johnson, and A. Aspuru-Guzik. Expressibility and entangling capability of parameterized quantum circuits for hybrid quantum-classical algorithms. *Advanced Quantum Technologies*, 2(12):1900070, 2019.

[243] D. R. Simon. On the power of quantum computation. In *Annual Symposium on Foundations of Computer Science*, 1994.

[244] V. Sitzmann, J. N. Martel, A. W. Bergman, D. B. Lindell, and G. Wetzstein. Implicit neural representations with periodic activation functions. In *Proc. NeurIPS*, 2020.

[245] J. Solomon, G. Peyré, V. G. Kim, and S. Sra. Entropic metric alignment for correspondence problems. *ACM Trans. Graph.*, 35(4), 2016.

[246] S. Speziali, F. Bianchi, A. Marini, L. Menculini, M. Proietti, L. F. Termite, A. Garinei, M. Marconi, and A. Delogu. Solving sensor placement problems in real water distribution networks using adiabatic quantum computation. *Quantum Computing and Engineering (QCE)*, 2021.

[247] T. Stollenwerk, E. Lobe, and M. Jung. Flight gate assignment with a quantum annealer. In *International Workshop on Quantum Technology and Optimization Problems*. Springer, 2019.

[248] B. Sun, A. Iliyasu, F. Yan, F. Dong, and K. Hirota. An rgb multi-channel representation for images on quantum computers. *Journal of Advanced Computational Intelligence and Intelligent Informatics*, 17:404–417, 2013.

[249] J. Sun, M. Ovsjanikov, and L. Guibas. A concise and provably informative multi-scale signature-based on heat diffusion. *Computer Graphics Forum*, 2009.

[250] R. Sundararaman, G. Pai, and M. Ovsjanikov. Implicit field supervision for robust non-rigid shape matching. In *Eur. Conf. Comput. Vis.*, 2022.

[251] P. Swoboda, C. Rother, H. Abu Alhaija, D. Kainmuller, and B. Savchynskyy. A study of lagrangean decompositions and dual ascent solvers for graph matching. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2017.

[252] A. Teplukhin, B. K. Kendrick, and D. Babikov. Calculation of molecular vibrational spectra on a quantum annealer. *Journal of chemical theory and computation*, 2019.

[253] P. Thai, M. T. Thai, T. Vu, and T. N. Dinh. Fasthare: fast hamiltonian reduction for large-scale quantum annealing. In *2022 IEEE International Conference on Quantum Computing and Engineering (QCE)*. IEEE, 2022.

[254] . ThinkLab. pygmtools. Shanghai Jiao Tong University. Github, 2021.

[255] F. Tombari, S. Salti, and L. Di Stefano. Unique signatures of histograms for local surface description. In *European Conference on Computer Vision (ECCV)*, 2010.

[256] L. Torresani, V. Kolmogorov, and C. Rother. Feature correspondence via graph matching: Models and global optimization. In *European conference on computer vision*, pages 596–609. Springer, 2008.

[257] L. Torresani, V. Kolmogorov, and C. Rother. A dual decomposition approach to feature correspondence. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 35(2):259–271, 2013.

[258] Y. Tsin and T. Kanade. A correlation-based approach to robust point set registration. In *European Conference on Computer Vision (ECCV)*, 2004.

[259] W. Van Dam, M. Mosca, and U. Vazirani. How powerful is adiabatic quantum computation? In *Proceedings 42nd IEEE symposium on foundations of computer science*, pages 279–287. IEEE, 2001.

[260] W. van Dam and G. Seroussi. Efficient quantum algorithms for estimating gauss sums. *arXiv preprint arXiv:quant-ph/0207131*, 2002.

[261] O. Van Kaick, H. Zhang, G. Hamarneh, and D. Cohen-Or. A survey on shape correspondence. In *Computer graphics forum*, volume 30, pages 1681–1707. Wiley Online Library, 2011.

[262] P. J. Van Laarhoven and E. H. Aarts. Simulated annealing. In *Simulated annealing: Theory and applications*, pages 7–15. Springer, 1987.

[263] S. E. Venegas-Andraca and S. Bose. Storing, processing, and retrieving an image using quantum mechanics. In *Quantum Information and Computation*, volume 5105, pages 137 – 147, 2003.

[264] G. Verdon, J. Pye, and M. Broughton. A universal training algorithm for quantum deep learning. In *APS March Meeting Abstracts*, 2019.

[265] M. Vestner, Z. Lähner, A. Boyarski, O. Litany, R. Slossberg, T. Remez, E. Rodola, A. Bronstein, M. Bronstein, R. Kimmel, and D. Cremers. Efficient deformable shape correspondence via kernel matching. In *Conference on 3D Vision (3DV)*, 2017.

[266] M. Vlastelica, A. Paulus, V. Musil, G. Martius, and M. Rolínek. Differentiation of blackbox combinatorial solvers. In *International Conference on Learning Representations (ICLR)*, 2020.

[267] N. Wiebe, D. Braun, and S. Lloyd. Quantum algorithm for data fitting. *Phys. Rev. Lett.*, 109:050505, 2012.

[268] D. Willsch, M. Willsch, H. De Raedt, and K. Michielsen. Support vector machines on the d-wave quantum annealer. *Computer Physics Communications*, 248:107006, 2020.

[269] B. Wu and B. Ghanem. ℓ_p-box admm: A versatile framework for integer programming. *IEEE transactions on pattern analysis and machine intelligence*, 41(7), 2018.

[270] Z. Wu, S. Song, A. Khosla, F. Yu, L. Zhang, X. Tang, and J. Xiao. 3d shapenets: A deep representation for volumetric shapes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1912–1920, 2015.

[271] X. Xu, L.-F. Cheong, and Z. Li. 3d rigid motion segmentation with mixed and unknown number of models. *IEEE Trans. Pattern Anal. Mach. Intell.*, 2019.

[272] F. Yan, A. M. Iliyasu, and S. E. Venegas-Andraca. A survey of quantum image representations. *Quantum Information Processing*, 15(1), 2016.

[273] F. Yan and S. E. Venegas-Andraca. Quantum image processing.

[274] A. Yurtsever, T. Birdal, and V. Golyanik. Q-fw: A hybrid classical-quantum frank-wolfe for quadratic binary optimization. In *European Conference on Computer Vision (ECCV)*, 2022.

[275] J.-N. Zaech, A. Liniger, M. Danelljan, D. Dai, and L. Van Gool. Adiabatic quantum computing for multi object tracking. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2022.

[276] M. Zaslavskiy, F. Bach, and J. Vert. A path following algorithm for the graph matching problem. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 31(12):2227–2242, 2009.

[277] Q. Zhao, S. Karisch, F. Rendl, and H. Wolkowicz. Semidefinite programming relaxations for the quadratic assignment problem. *Journal of Combinatorial Optimization*, 2, 10 1997.

[278] F. Zhou and F. De la Torre. Factorized graph matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 38(9):1774–1789, 2016.

[279] K. Zick, O. Shehab, and M. French. Experimental quantum annealing: Case study involving the graph isomorphism problem. *Scientific Reports*, 5, 2015.

[280] S. Zuffi, A. Kanazawa, D. Jacobs, and M. J. Black. 3D menagerie: Modeling the 3D shape and pose of animals. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2017.