Q-Seg: Quantum Annealing-Based Unsupervised Image Segmentation

Supreeth Mysore Venkatesh, *German Research Center for Artificial Intelligence (DFKI), Saarbruecken, Germany; University of Kaiserslautern-Landau (RPTU), Kaiserslautern, Germany*

Antonio Macaluso, *German Research Center for Artificial Intelligence (DFKI), Saarbruecken, Germany*

Marlon Nuske, *German Research Center for Artificial Intelligence (DFKI), Kaiserslautern, Germany*

Matthias Klusch, *German Research Center for Artificial Intelligence (DFKI), Saarbruecken, Germany*

Andreas Dengel, *German Research Center for Artificial Intelligence (DFKI), Kaiserslautern, Germany; University of Kaiserslautern-Landau (RPTU), Kaiserslautern, Germany*

Abstract—We present Q-Seg, a novel unsupervised image segmentation method based on quantum annealing, tailored for existing quantum hardware. We formulate the pixel-wise segmentation problem, which assimilates spectral and spatial information of the image, as a graph-cut optimization task. Our method efficiently leverages the interconnected qubit topology of the D-Wave Advantage device, offering superior scalability over existing quantum approaches and outperforming several tested state-of-the-art classical methods. Empirical evaluations on synthetic datasets have shown that Q-Seg has better runtime performance than the state-of-the-art classical optimizer Gurobi. The method has also been tested on earth observation image segmentation, a critical area with noisy and unreliable annotations. In the era of noisy intermediate-scale quantum, Q-Seg emerges as a reliable contender for real-world applications in comparison to advanced techniques like Segment Anything. Consequently, Q-Seg offers a promising solution using available quantum hardware, especially in situations constrained by limited labeled data and the need for efficient computational runtime.

Image segmentation is pivotal in computer vision, enabling pattern recognition across diverse applications including medical imaging, autonomous navigation, search engines, and earth observation. Technically, segmenting an image consists of partitioning it into non-overlapping regions with distinct attributes, aligning with human perception. Algorithmic advancements in computer vision research, leading to substantial performance enhancements, are closely linked to underlying hardware capabilities.

Recently, quantum computing has emerged as a powerful alternative to solve complex real-world applications. However, current quantum computers have limited capabilities, impacting algorithm design and testing. Specifically for computer vision, several quantum methods have been proposed for image classification, multi-object tracking, image matching, motion segmentation, edge detection and image segmentation. Nonetheless, all these approaches do not show a concrete advantage for practical applications using existing quantum hardware.

In this paper, we introduce *Q-Seg*, a novel unsupervised method that formulates the image segmentation problem as a graph-based optimization task, solvable via quantum annealing. The vast solution space, exponential in pixel count, is efficiently navigated to yield near-optimal segmentation congruent with contemporary annealer architectures. To show the ready-to-use capability of our approach, we run experiments using the *D-Wave Advantage* annealer on synthetic and real-world satellite imagery datasets, specifically for Forest Cover and Flood Mapping. In the case of Forest Cover segmentation, we obtain high-quality masks, underscor-
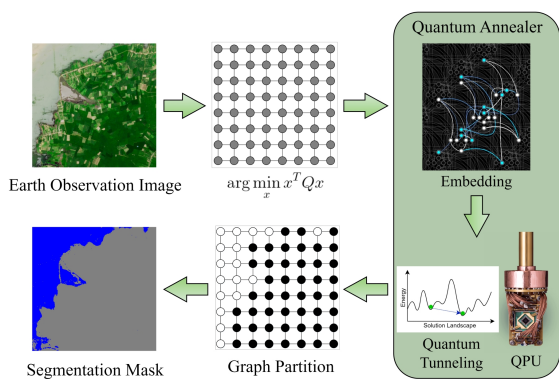
FIGURE 1. The proposed *Q-Seg* for unsupervised image segmentation generates a graph representing the original image. The distinct semantic regions in the image are identified by finding the minimum cut in the graph, an NP-hard problem that can be formulated as Quadratic Unconstrained Binary Optimization (QUBO). The QUBO problem is subsequently encoded into the physical topology of the D-Wave quantum annealer. Leveraging quantum tunneling, the annealing process efficiently explores an exponentially large solution space to locate the global optimum, which can be decoded as the segmentation mask.

ing the utility of unsupervised methods in scenarios with noisy or uncertain ground truth. For Flood Mapping, we achieve near-optimal segmentation quality, rivaling supervised methods. These results accentuate the capability of *Q-Seg* to segment small and medium-sized images with existing quantum annealers, benchmarked against the classical state-of-the-art optimizer *Gurobi* for runtime and quality assessment and also compared against *Segment Anything Model (SAM)* on real-world images.

RELATED WORKS

State-of-the-art methods for image segmentation primarily involve the use of Deep Convolutional Neural Networks in a supervised manner. These approaches can achieve remarkable segmentation performance when supplied with accurate annotations [1]. However, in various domains acquiring labeled data encounters challenges such as the considerable cost and time required for manual annotation, the necessity for domain expertise, and scalability issues when handling large datasets. Also, subjectivity among annotators and variations in interpretation can introduce inconsistencies.

These limitations underscore the importance of unsupervised techniques, especially when leveraging vast unlabeled datasets [2]. Unsupervised segmentation algorithms can be classified into several categories.

Thresholding is a pixel-wise segmentation method that assigns pixels to regions based on intensity values. While computationally efficient, it is susceptible to issues arising from illumination variations and noise, which reduce its potency in situations where factors beyond intensity affect performance [3]. On the other hand, feature space clustering segments the spectral feature space into distinct clusters, each representing a region. However, a notable limitation is its disregard for spatial relationships between pixels [4]. Edge-based segmentation techniques detect discontinuities in image attributes, primarily pixel intensities, to delineate object boundaries. While adept at handling images with clear object edges, this method struggles with noisy or indistinct boundaries, resulting in segmentation errors. Region-based methods focus on pixel homogeneity. Techniques like region-growing initialize with seed pixels and expand regions based on similarity metrics until a set condition is satisfied. Conversely, region-splitting dissects larger segments into finer clusters when they lack homogeneity in attributes like color, texture, or intensity [5]. However, region-based methods falter in scenarios where objects exhibit similar attributes or when similarity thresholds are not optimally set, resulting in over or under-segmentation [6].

Graph-based segmentation conceptualizes image segmentation as a graph partitioning task where nodes correspond to pixels or regions, while edges, weighted by pixel dissimilarity, link adjacent nodes [7]. The primary goal is to discern subgraphs that align with coherent clusters in the image. A salient feature of this approach is its capacity to amalgamate both boundary and regional information, yielding globally optimal solutions for specific cases. This makes it a potent tool for image segmentation, especially when prior data is sparse. However, the inherent complexity of optimization tasks, such as graph cuts or clustering, renders the method computationally demanding [8].

Research has advanced beyond classical computing paradigms, predominantly exploring quantum algorithms for computer vision tasks [9]. Graph-based segmentation using variational quantum circuits has been investigated [10]. However, the method has scalability issues in practical applications, as the qubit requirements, corresponding to the number of image pixels, limit their feasibility for large-scale use.

A recent study [11] explores the use of quantum annealing for image segmentation by incorporating the Expectation Maximization (EM) algorithm into graph-based segmentation as a constrained optimization problem. This approach, while innovative, does not ensure that solutions to the reformulated unconstrained problem align with those of the initial constrained problem [12],

and faces scalability issues due to the graph structure. Furthermore, the comparison of quantum annealer performance to classical methods lacks engagement with advanced segmentation techniques. Conversely, *Q-Seg* proposes a simplified graph structure, starting directly with an unconstrained optimization problem. It eliminates redundant solutions, offering a completely unsupervised approach, enhancing scalability and efficiency. Our experiments detail a thorough analysis of the physical qubit requirements, affirming *Q-Seg*'s advancements.

PRELIMINARIES

In contemporary quantum computing, two distinct paradigms have broadly emerged for tackling diverse problem sets: universal gate quantum computing and adiabatic quantum computing (AQC). This paper focuses on the adiabatic quantum computing paradigm, a computational approach that employs the quantum annealing process to address quadratic unconstrained binary optimization (QUBO) problems, defined as follows [13]:

$$\arg\min_x \; x^T Q x = \arg\min_x \sum_{i=1}^{n} l_i x_i + \sum_{1 \le i < j \le n} q_{ij} x_i x_j \quad (1)$$

where $x \in \{0,1\}^n$ is the binary variable vector and $Q \in \mathbf{R}^{n \times n}$ is the QUBO matrix, with its diagonal and off-diagonal elements representing linear (l_i) and quadratic (q_{ij}) coefficients, respectively. Despite its straightforward formulation, the QUBO problem is NP-hard, presenting significant computational challenges for large-scale instances since the solution space grows exponentially with the input size [13]. Moreover, unlike continuous optimization, the discrete optimization landscape here demonstrates non-linear cost deviations, i.e., small changes in input can lead to large variations in the cost function.

Quantum annealers, utilizing the principles of quantum mechanics, particularly the adiabatic theorem, efficiently navigate complex solution landscapes. Through quantum tunneling, these devices can potentially escape local minima, offering a promising alternative to classical optimization methods. However, the practical implementation of large QUBO problems on quantum annealers is hindered by limitations such as a finite number of qubits, connectivity constraints, susceptibility to noise, and sensitivity to optimization landscape and algorithmic characteristics. Thus, identifying suitable problems is essential for unlocking the full potential of quantum annealing in tackling complex optimization tasks.

METHODS

This section describes *Q-Seg* in detail, delineating the algorithmic workflow, encapsulating the preprocessing, problem formulation, quantum annealing process, and how to obtain the segmentation mask.

Image to Graph: In graph-based segmentation, a graph is constructed with a node for each image pixel, typically supplemented by two terminal nodes termed *source* and *sink*, which serve as theoretical constructs for the object and the background. Edges linking adjacent nodes incorporate weights that reflect either similarity or dissimilarity. Additionally, the source and sink nodes connect to all pixel-representing nodes, with edge weights encoding prior knowledge. Evolving from conventional techniques, our strategy exploits the quantum annealer hardware's potential, presenting a formulation that achieves greater efficiency relative to classical counterparts. *Q-Seg* begins by forming a lattice graph from the input image, adopting a grid-graph structure and omitting the terminal nodes. In this representation, graph nodes equate to image pixels, preserving the spatial inter-pixel connectivity crucial for segmentation. The choice of defining the edge weights is application-dependent, and different usecases may require distinct metrics tailored to specific characteristics of the image data. For scenarios with limited annotated samples, establishing optimal edge weights is approached as an optimization problem, detailed further in the appendix.

In our approach, edge weights represent the similarity between neighboring pixels' intensity values, calculated using a Gaussian similarity metric. For neighboring pixels p_i and p_j with intensity values $I(p_i)$ and $I(p_j)$, respectively, we use the following function to compute the edge weight [14]:

$$w'(p_i, p_j) = 1 - \exp\left(-\frac{(I(p_i) - I(p_j))^2}{2\sigma^2}\right) \quad (2)$$

where σ is the standard deviation parameter that controls the spread of the Gaussian function. This similarity metric yields values in the range $[0,1]$.

To ensure the edge weights are both positive and negative, we normalize these weights to the range $[-1, 1]$. This is done by first finding the minimum and maximum weights across all edges, and then applying the following normalization formula:

$$w(p_i, p_j) = -1 \times \left(\frac{(b - a) \cdot (w'(p_i, p_j) - \min(w))}{\max(w) - \min(w)} + a \right) \quad (3)$$

where $a = -1$ and $b = 1$ are the desired range bounds. Normalizing the edge weights to include both positive

and negative values is crucial as it ensures that the segmentation algorithm can effectively differentiate between similar and dissimilar pixel pairs. This formulation of the edge weights provides a general similarity score that effectively works for the experiments in this paper, but every specific use case might require a custom edge weight metric.

When representing an image as a graph, segmentation involves identifying a graph cut that partitions the vertices based on the similarity metric of edge weights. Specifically, when the edge weights represent pixel similarity, dividing the graph into semantically distinct regions can be framed as discovering the minimum cut in the graph, minimizing the sum of cut edges' overall similarity. Conversely, when edge weights measure pixel dissimilarity, the segmentation task targets the maximum cut. For real-valued edge weights, the maximum cut problem can be transformed into a minimum cut problem by negating the edge weights and vice-versa. Consider an undirected weighted graph $G(V, w)$ with no self-loop edges, where V is the set of vertices and $w : V \times V \to \mathbf{R}$ is the edge weight function. In this context, a 'cut' denotes the partitioning of the graph into disconnected subgraphs, which arises from the removal of a specific set of edges. The *cost* of a cut is quantified as the aggregate of the weights of the edges removed as follows:

$$\mathcal{MINCUT}(G) = \arg \min_{A,\overline{A}} \sum_{i \in A, j \in \overline{A}} w(v_i, v_j) \qquad (4)$$

$$\text{such that } A \cup \overline{A} = V, A \cap \overline{A} = \varnothing$$

where $w(v_i, v_j)$ is the weight of the edge connecting the nodes v_i and v_j. In its general formulation where edge weights can be positive and negative, finding the minimum (or maximum) cut in the graph is NP-Hard. However, there are heuristic algorithms for finding minimum cuts in planar graphs in polynomial time for specific instances [15]. Nevertheless, for the generic case, the problem still remains NP-complete [16]. Indeed, for an image with n pixels, there exist 2^n potential solutions, rendering exhaustive enumeration of all possible solutions infeasible. Fig. 2 illustrates converting a grey image of size 3×3 to a grid graph where the edge weights $\in [-1, 1]$ represent the similarity between the neighboring pixels, as described by Eq. 3.

Formulating QUBO: To overcome the computational hurdles on a classical computer, and be able to segment an image efficiently, we use a quantum annealer. This necessitates recasting the minimum cut problem into QUBO form, which is compatible with the operational framework of quantum annealing. A cut is also conceptualized as separating the vertex
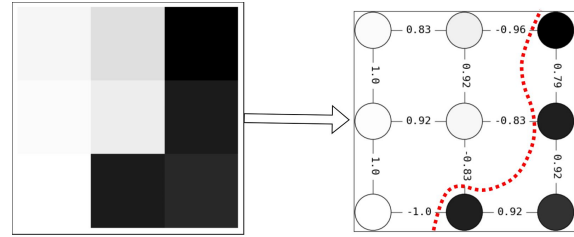
FIGURE 2. Converting the pixel values into edge weights in the grid graph. The grid structure of the graph captures the spatial information (high-level information such as regions or objects) and the edge weight captures the spectral information (low-level information such as pixel values). The red-colored dotted curve passes through the set of edges that divide the vertices into two distinct sets, such that the sum of the edge weights is minimum.

set into two exclusive subsets, with a minimum cut characterized by the least possible sum of edge weights bridging these subsets. If G has n nodes, associating a binary variable x_{v_i} to vertex v_i for all $i \in \{1, 2, ..., n\}$ allows to encode the solution as $v_i \in A \,\forall\, x_{v_i} = 0$ and $v_i \in \overline{A} \,\forall\, x_{v_i} = 1$ or vise-versa. The objective function for computing the minimum cut in this binary framework is articulated as [13]:

$$x^* = \arg \min_x \sum_{1 \le i < j \le n} x_{v_i}(1 - x_{v_j})w(v_i, v_j) \qquad (5)$$

$$\text{where } x = x_{v_1}, x_{v_2}, ..., x_{v_n} \text{and } x_{v_i} \in \{0, 1\}$$

The cost function in Eq. (5) is equivalent to the QUBO formulation in Eq. (1) [17]. Given the edge weights $w(v_i, v_j)$, substituting them into Eq. (5) and simplifying yields a quadratic expression in binary variables x_{v_i} where $i \in \{1, 2, ..., n\}$. The construction of the Q matrix in Eq.(1), is derived from the coefficients of terms in the quadratic equation. Note that the QUBO formulation of the minimum cut does not introduce any overhead in terms of the number of binary variables compared to the input size in the original problem, as is the case with other existing formulations [18], [11].

QUBO with Quantum Annealing: Quantum annealing offers a probabilistic approach to solving the QUBO problem through the exploitation of quantum tunneling and entanglement. The process begins by mapping the QUBO onto the topology of physical qubits, aligning the solution with the binary states of the qubits. Throughout the annealing process, the system evolves from a superposition of all possible states toward the lowest energy state, corresponding to a state where the optimal QUBO solution has a high probability of being measured. This evolution is governed by the annealing

schedule, which carefully balances quantum properties and classical interactions to avoid local minima.

The *D-Wave* quantum annealer, in particular, employs a specialized hardware design conducive to this process, making it well-suited for graph-based problems like image segmentation. As an annealer is a probabilistic machine, typically an execution involves several runs, and it samples a possible solution at each run. Through iterative adjustments of the system's parameters and repeated annealing cycles, the probability of discovering the global minimum increases. The final readout produces a binary string directly translating into the segmented image.

To summarize, the *Q-Seg* approach is described in Fig. 1, and the pseudo-code is shown in Algorithm 1.

Algorithm 1 Outline of *Q-Seg*

Require: Image I with n pixels
Ensure: Segmentation mask M matching I's dimensions
  1: Construct grid graph $G(V, w)$ from image I
  2: Formulate QUBO for minimum cut on G (Eq. (5))
  3: Map the QUBO to the quantum annealer's architecture
  4: Run the quantum annealing
  5: Extract the lowest-energy sample X^*
  6: Decode sample X^* into segmentation mask M

For an input image I, we construct a grid graph G with a one-to-one correspondence between pixels in I and vertices in G. Edge weights $w(v_i, v_j)$ are assigned based on pixel similarity between v_i and v_j, to segment I by identifying a minimum cut in G. The QUBO formulation captures this objective, translating the problem into a format amenable to quantum annealing. The annealing process is iterated multiple times to ensure the reliability of the solution, which is then decoded into a binary mask M as $M_i = x_{v_i}$ for each vertex $v_i \in V$, describing the regions in I.

The structure of the graph in our problem formulation, with nodes exhibiting a maximum degree of 4, naturally fits the architecture of the existing quantum annealer hardware. This congruence facilitates efficient mapping of logical to physical qubits, reducing the number of cloned qubits and inter-qubit coupling.

EXPERIMENTS

In this section, we present experimental evaluations of *Q-Seg* against state-of-the-art classical methods on a synthetic dataset and two practical usecases in the domain of earth observation.

Datasets

Synthetic Data: The complex operation in *Q-Seg* involves computing the minimum cut in a grid graph, a key step essential for effective image segmentation. We generate synthetic datasets comprising undirected, connected, and weighted grid graphs of square structure, with sizes ranging from 2×2 up to 44×44. This data allows assessing the performance of *Q-Seg* in terms of runtime while increasing the problem size. For reproducibility, we employ a fixed seed to assign edge weights, which are randomly selected from a uniform distribution within the interval $[-1, 1]$ ensuring that the edge weights are both, positive and negative. Tests are conducted on five sets of graphs, each initialized with a different seed.

Forest Cover: Forest segmentation is crucial for ecological sustainability enabling accurate assessment of deforestation rates, forest density, and changes due to climatic or human factors. Due to the limited availability of time resources of the real quantum device, we randomly sample a subset of images from the *DeepGlobe18* dataset[1], each measuring 256×256 pixels, accompanied by binary segmentation mask. Furthermore, the original images are downscaled to 32×32 pixels, to fit the actual number of qubits in the real device. In particular, the preprocessing steps included median blurring and conversion to the HSV (Hue Saturation Value) color scale, tailored to the specific usecase. For generating the input graph for *Q-Seg* algorithm, we incorporate the Gaussian similarity (Eq. 2) as the edge weight metric, normalizing them within the range of $[-1, 1]$ (Eq. 3).

Flood Mapping: The *Sen1Floods11* dataset[2], designed for water segmentation in satellite imagery, comprises georeferenced samples with dimensions of 512×512 pixels across 13 spectral bands. The ground truth masks include binary labels and an additional value of -1, indicating uncertain areas, as annotated by experts. Specifically, the subset of images used comprises pictures of the Bolivia region obtained from [19]. The images are segmented at a pixel level, and then we work with 32×32 patches and subsequently integrate these patches to form the complete segmentation mask. This approach allowed for detailed analysis without compromising image integrity.

As preprocessing and customizing input for the subsequent task is typical for an unsupervised algorithm when applying it to a real-world usecase, likewise we

---

[1] https://www.kaggle.com/datasets/quadeer15sh/augmented-forest-segmentation
[2] https://github.com/cloudtostreet/Sen1Floods11

follow a few steps for generating the graph for *Q-Seg*. We first calculated the normalized difference water index of the input images and used a normalized Gaussian similarity metric as discussed in Eq. 3.

Solving methods

Our experiments were conducted using the *D-Wave Advantage* quantum annealer for the graph-cut optimization, accessed through a cloud service via the *dimod* [3] library. This device comprises 5670 physical qubits and has *pegasus* topology. We evaluated *Q-Seg* against the state-of-the-art solver *Gurobi*. For benchmarking against classical approaches, we implemented the *Gurobi v9.5.2* optimizer to solve the QUBO problem representing the minimum cut. Furthermore, to quantitatively assess the segmentation quality, we compare the results of *Q-Seg* for the Flood Mapping dataset with a state-of-the-art supervised method that uses gradient-boosted decision trees (*GBDT*) [19]. The classical part of our experiments was conducted on a system equipped with an *Intel(R) Xeon(R)* CPU @ *2.20GHz* and *8 GB* RAM, with all software developed in *Python 3.9*.

For real-world datasets, we also compare our results with the recent State-of-the-Art segmentation method *SAM*, known for its robust zero-shot capabilities across various domains [20]. To maintain a fair comparison with our unsupervised method, we employed the latest version of *ViT-H*[4] model pretrained on *SA-1B* [5] dataset containing 11 million images and 1.1 billion masks, without specifically fine-tuning the model weights for the target usecase. We allowed *SAM* to generate 3 segmentation masks for each image and the one with the highest confidence score is illustrated in the results.

In our experiments, we explored an alternative approach to solving the minimum cut problem with *Gurobi*, adopting a constrained optimization mode. This approach, focusing on edge encoding, increases variable count significantly and resulted in poorer performance. Furthermore, we considered comparing our quantum annealing results with simulated annealing, often regarded as its direct competitor. However, research indicates simulated annealing falls short of quantum annealing in both speed and solution quality, often by exponential margins, leading us to omit it from our comparative analysis.

Finally, we omit the experimental comparison with

the recently introduced segmentation method using quantum annealers [11] for three reasons. First, the method works on pivotal assumptions for implementing the EM algorithm and requires prior knowledge about the distribution of pixel values of different regions, making it not fully unsupervised. Second, the increased overhead in terms of the number of physical qubits hinders the possibility of running larger problem instances as those implemented to test *Q-Seg*. Third, the code is not made publicly available. Nevertheless, we present a scalability analysis based on the number of physical qubits required by the existing quantum hardware and compare it with that of *Q-Seg* (see Fig. 9).

Metrics

We evaluate our method using three different types of metrics: *runtime*, *quality* and *scalability*.

The *runtime* allows us to estimate the efficiency of *Q-Seg* in comparison to classical state-of-the-art optimizers. In Fig. 3, a comprehensive description of all components of the quantum annealing pipeline is presented, emphasizing their significance when solving a problem on *D-Wave* annealers that is accessible via the cloud.

In terms of *quality*, we evaluate *Q-Seg* using a set of established metrics. These include *Intersection over Union (IoU)* to assess the accuracy of overlap in segmented regions, *Accuracy* for determining the correct segmentation of pixels, *Recall*, and *Precision* to evaluate the detection and accuracy of segmentation.

Additionally, to assess the segmentation carried out by *Q-Seg* in comparison to deterministic methods, we conducted a comparison of minimum cut values, employing the relative error as the metric for evaluation, defined as:

$$E_r = \left| \frac{V_G - V_{QA}}{V_G} \right| \qquad (6)$$

where V_G represents the minimum cut value from *Gurobi* and V_{QA} from the *D-Wave Advantage* annealer.

Finally, we assess the *scalability* of *Q-Seg* by quantifying *Qubit Complexity*, i.e., the number of physical qubits required in a real quantum device with respect to other existing quantum-annealing-based approaches.
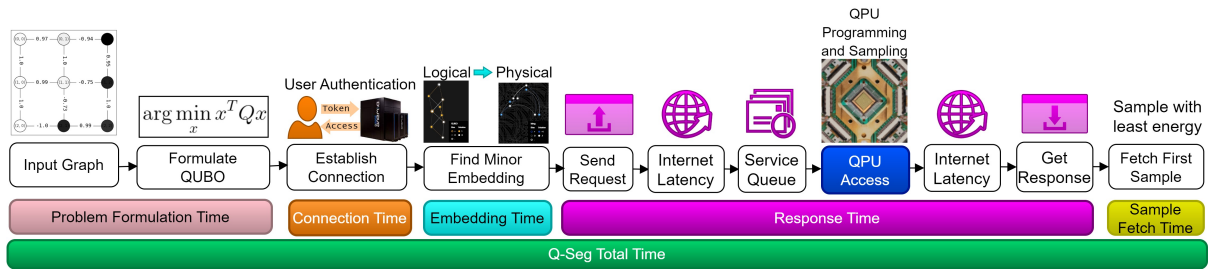
RESULTS

Runtime Analysis

We executed *Q-Seg* on the synthetic dataset using both the *D-Wave Advantage* and *Gurobi* to compare classical and quantum algorithms while progressively increasing the problem size. The results, which encompass all the distinct components in the quantum annealing pipeline (Fig. 3), are illustrated in Fig. 4. We observe that

---

[3]https://docs.ocean.dwavesys.com/en/stable/docs_dimod/
[4]https://github.com/facebookresearch/segment-anything.
[5]https://ai.meta.com/datasets/segment-anything-downloads/

FIGURE 3. The figure illustrates the operational pipeline of *D-Wave* quantum annealer. It begins with reformulating the minimum cut problem as a QUBO, followed by authentication using a private token for remote access. The *minorminer* tool maps the logical qubits in QUBO to the physical qubits in hardware. The problem instance is sent over the internet to join the queue on the shared *D-Wave* device. The Quantum Processing Unit (QPU) performs the annealing process, producing a set of samples. The final step is extracting the optimal solution, identified by its lowest energy state that encodes the segmentation mask.
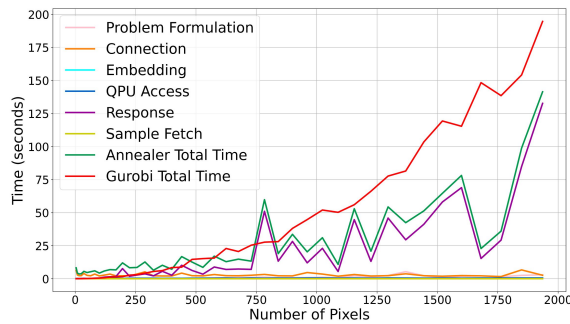

FIGURE 4. Runtime comparison between *D-Wave Advantage* annealer and *Gurobi* for the synthetic data with *seed* = 333. This graph illustrates the breakdown of annealer runtime, highlighting the components in Fig. 3. Despite shared and remote access, the *D-Wave Advantage* demonstrates a consistently shorter total runtime compared to *Gurobi*'s local execution.

even when factoring in the runtime, which includes the response time from sending the problem to receiving the solution from the *D-Wave* annealer—involving internet latency, queuing due to shared annealer usage, and the annealing process—the total time-to-solution for quantum annealing is significantly lower than that of *Gurobi* that executes locally.

In a hypothetical scenario with a dedicated QPU, we could omit the connection establishment, internet latency, and queuing, which are significant contributors to the total runtime. This assumption allows for a more direct comparison of the annealer's efficiency against *Gurobi*. In particular, *QPU Access Time* reflects the Quantum Processing Unit's active problem-solving duration. This metric can be divided into *QPU Programming Time*, *QPU Sampling Time*, and *QPU Access Overhead*

Time. The programming phase sets up the QUBO model in the annealer, sampling involves the actual annealing process, and the overhead time is associated with post-processing the samples.
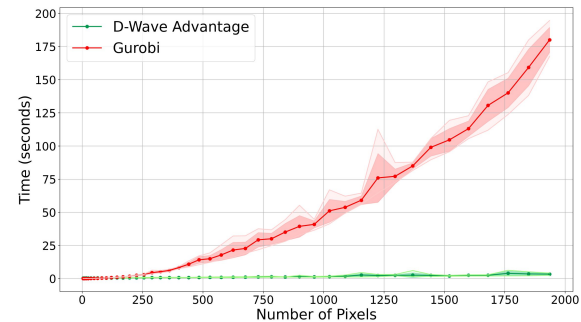

FIGURE 5. The plot illustrates the mean runtime (represented by a solid line), the range of runtimes (indicated by the broadly shaded area), and the standard deviation (denoted by the lightly shaded area) is aggregated over five sets of synthetic data for the minimum cut operation on graphs of square images of varying sizes. The results highlight the efficiency of *Q-Seg*, especially evident in processing larger images.

Fig. 5 compares the runtime of *D-Wave Advantage* and *Gurobi* assuming a dedicated onsite QPU. The hybrid process, involving QUBO formulation, embedding, annealing, and solution retrieval, exhibits even more pronounced superior performance compared to the previous scenario, especially as the problem size increases. This efficiency can be attributed to the grid graph structure aligning well with the QPU architecture. The newer *D-Wave* devices, featuring enhanced qubit connectivity, further enhance this efficiency, making *Q-Seg* particularly effective on these platforms.

Quality

While *D-Wave* exhibits faster processing than *Gurobi*, it is essential to evaluate the quality of the solutions. The analysis depicted in Fig. 6 shows that although *Gurobi* consistently produces superior solutions, the margin is minimal. This observation is further supported by the relative error trends across different problem sizes.
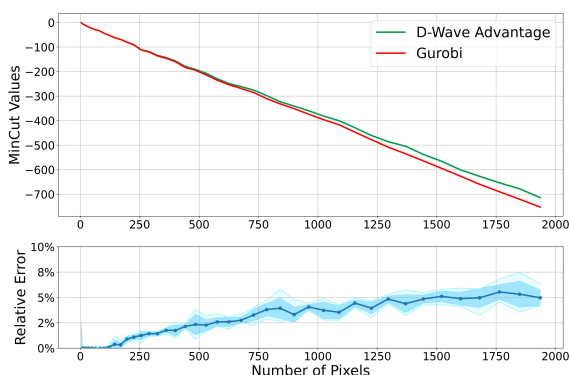

FIGURE 6. The first plot contrasts the minimum cut values obtained by *Gurobi* and the *D-Wave Advantage* annealer on synthetic data with *seed* = 333, highlighting *Gurobi*'s marginally better performance. The trend of diminishing values substantiates the rising count of negative edges that are cut as the problem size increases. To substantiate a consistent behavior, the second plot shows the mean (represented by the solid line), range (depicted by the broadly shaded area) and standard deviation (illustrated by the narrow shaded area) of the relative errors in the annealer solutions compared to *Gurobi* aggregated over five sets of the synthetic data.

Our approach demonstrates the ability to address large-scale problems with commendable quality using current quantum hardware. With an onsite annealer, the limitations on the number of annealing cycles, impacting the quality of solutions, could be mitigated, potentially further elevating the solution quality.

Forest Cover

Segmenting Forest Cover poses a significant challenge, primarily attributed to the presence of noisy and inconsistent ground truth labels in the dataset. This inconsistency can introduce bias during the training of supervised methods, potentially leading to misleading comparisons between algorithmic outcomes and ground truth masks. In addressing these challenges, we compared the results of *Q-Seg* with that of *SAM*. Fig. 7 provides a qualitative glimpse into the segmentation results of *Q-Seg* against existing masks and contrasts these with the outcomes from the *SAM* model.

Even state-of-the-art methods like *SAM* fail to generalize for specific usecases, and even harder for
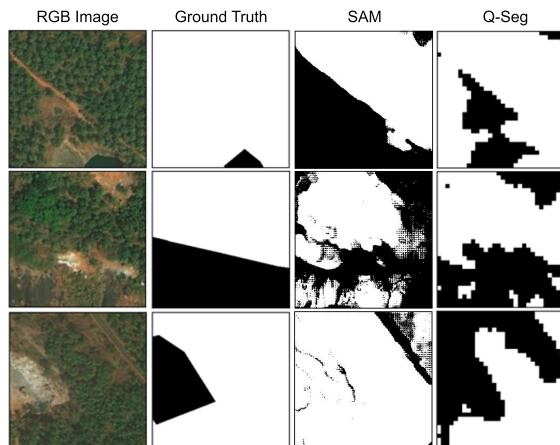
FIGURE 7. Visualization of RGB images, ground truth masks, *Q-Seg*, and *SAM* segmentation results in the Forest Cover usecase. The ground truth masks show noisy labeling and irregularities, while the *Q-Seg* results reveal more coherent and visually refined segmentation of Forest Cover, demonstrating notable improvements over *SAM* in handling noisy labels.

datasets with noisy labels. Our approach showcases its potential to provide more reliable and visually consistent segmentation results in such challenging environments. Importantly, the consistency of poor-quality ground truth and the corresponding high-quality masks provided by *Q-Seg* is congruous across most images in the dataset[6]. By preprocessing the image adapting to the usecase and using a suitable edge weight metric, *Q-Seg* emerges as a better contender for unseen images when compared to existing state-of-the-art techniques, including *SAM*. Since the Forest Cover dataset is not well labeled, a quantitative comparison of the results of *Q-Seg* with the noisy labels would not be a fair measure and thus we extend our experiments with the Flood Mapping usecase. As a takeaway, the results demonstrate the effectiveness of *Q-Seg* making graph-based techniques feasible by leveraging the capabilities of quantum annealers.
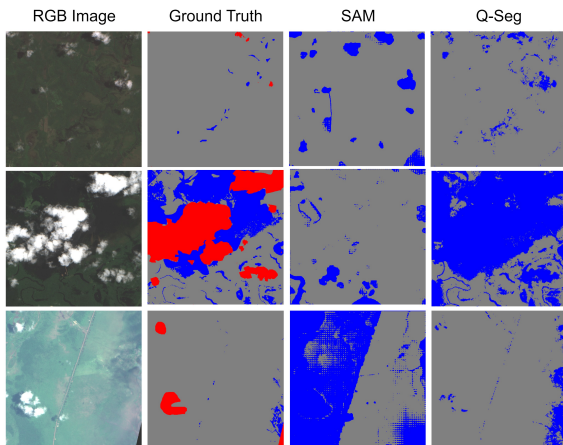
Flood Mapping

Commonly, state-of-the-art methods for image segmentation rely on supervised learning, utilizing annotated images to train algorithms for correctly identifying masks of unlabelled images. In the context of the flood mapping dataset, the *GBDT* stands out as the state-of-the-art method, even outperforming CNNs in this specific case [19]. Alongside *GBDT*, we include a

---

[6]https://github.com/supreethmv/Q-Seg

comparison with *SAM*, using its pretrained model to assess performance in this usecase.

To evaluate the performance of *Q-Seg* in terms of the quality of the segmentation, we compare the results with both *GBDT* and *SAM*. The output masks can be seen in Figure 8, and the quantitative results are reported in Table 1.

FIGURE 8. Visualization of RGB images, expert-labeled ground truth masks, *Q-Seg*, and *SAM* segmentation results in Flood Mapping. The segmentation differentiates water (blue), dry land (grey), and uncertain areas (red) affected by cloud coverage. *Q-Seg* is observed to be over-segmenting in comparison to the ground truth, which leads to higher precision and shows inconsistent behavior on clouds, suggesting possibilities for improvement in the preprocessing techniques.

TABLE 1. Performance Comparison of the supervised approach *GBDT* [19], *SAM*, and *Q-Seg* in Flood Mapping usecase. The metrics are evaluated ignoring the unsure pixels in the ground truth.

Metrics	GBDT [19]	SAM	Q-Seg
IoU	**0.888**	0.330	0.775
Accuracy	**0.913**	0.463	0.828
Recall	**0.993**	0.340	0.780
Precision	0.892	0.887	**0.990**
F1-score	**0.935**	0.393	0.861

Despite the latest and sophisticated capabilities of *SAM*, it demonstrates substandard performance on the unseen dataset, highlighting the robust behavior of *Q-Seg* when applying case-specific metrics. As expected, the supervised approach outperforms *Q-Seg* in the majority of metrics, except for *Precision* where *Q-Seg* exhibits better performance, likely due to its tendency to over-segment. Nevertheless, *Q-*

Seg delivers commendable results, closely trailing the supervised approach despite being an unsupervised method and showing more robust behavior than *SAM* in this context. Although *Q-Seg* cannot outperform *GBDT*, it could still be a desirable approach when the quality of the output happens to be tolerable for the downstream application. Moreover, *Q-Seg* circumvents the requirement of reliable hand-labeled data by domain experts and the extensive process of training a supervised learning model, showcasing its potential as a robust tool in challenging segmentation scenarios.

Scalability

In this section, we delve into the scalability of our method, particularly in the context of current quantum annealing technologies and their near-term advancements. Our focus is on assessing the adaptability of *Q-Seg* to the evolving landscape of quantum computing, especially in comparison to another quantum annealer-based image segmentation method used for Synthetic Aperture Radar (SAR) images [11]. A critical aspect of our analysis is the compatibility of *Q-Seg* with the topological structure of the *D-Wave Advantage* quantum annealers. Unlike the approach in [11], *Q-Seg* demonstrates a better utilization of the quantum hardware capabilities by efficient embedding of binary variables in the QUBO formulation to the physical qubits of the annealer hardware. As the structure of the graphs used in both approaches are known and *D-Wave* provides the couplers among the physical qubits available in the *Advantage* annealer, we can find the embedding that will be used for solving the QUBO. Thus, we can deduce the number of physical qubits and eventually reveal the scalability of these approaches.
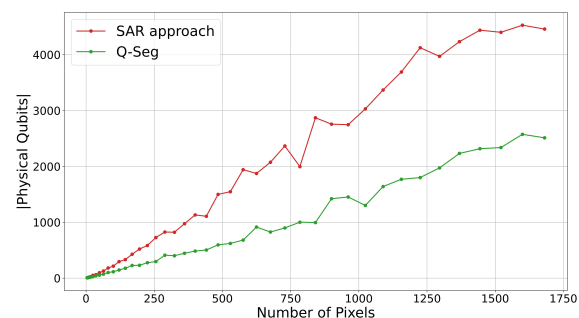
FIGURE 9. Comparison of physical qubit required for *Q-Seg* and SAR segmentation [11] on the *D-Wave Advantage* with varying image sizes. The qubit requirements are estimated using *minorminer* tool integrated into *dimod* library, given the logical qubit interactions (quadratic terms in QUBO problem) and the available couplers in the annealer hardware.

The results, as shown in Fig. 9, denote a substantial reduction in the physical qubits required for *Q-Seg* compared to the method in [11]. This efficiency arises from *Q-Seg*'s problem formulation: for an image with *n* pixels, it necessitates *n* vertices, each with a maximum degree of 4. In contrast, the method in [11] utilizes $n + 2$ vertices, with the terminal vertices having a degree of *n*. The edges in the graph represent logical qubit interactions, where assigning a logical qubit with extensive interactions to a physical qubit in the QPU poses challenges, requiring multiple physical qubit clones for effective representation. This directly impacts the annealer performance, as an ideal embedding involves a one-to-one mapping of logical to physical qubits. Furthermore, the solution space complexity for the method in [11] is 2^{n+2}, four times that of *Q-Seg*. Given that there are only up to 2^n possible binary segmentations of the image, all encompassed within the solution space of *Q-Seg*, it suggests a prevalence of redundant solutions in the method of [11].

CONCLUSION

In this paper, we introduced *Q-Seg*, a novel unsupervised segmentation algorithm designed specifically for existing quantum annealers. By efficiently reformulating the segmentation task as a minimum-cut problem in a graph and subsequently as a QUBO problem, our proposed quantum approach efficiently leverages the hardware topology of *D-Wave* annealers, harnessing the advantages of quantum computing. Along with enhancing the scalability, it ensures the generation of high-quality solutions using near-term quantum devices.

Our experiments on synthetic data with simulated images of varying sizes demonstrate a significant runtime advantage of *Q-Seg* in processing large-scale images compared to the state-of-the-art optimizer *Gurobi* (Fig. 5). Notably, the runtime efficacy is achieved without a significant impact on the quality (Fig. 6). Further, *Q-Seg* was applied to real-world tasks such as Forest Cover and Flood Mapping to test its applicability. The results suggest a potential for quantum computing to contribute to image segmentation, especially in scenarios with noisy and unreliable annotations. The results hint at *Q-Seg*'s potential as a practical tool for certain segmentation tasks without extensive training or expert-labeled data, while also acknowledging the need for further exploration and validation.

The future direction for *Q-Seg* is aimed at broadening its application scope beyond earth observation and enhancing segmentation quality. Upcoming enhancements include extending the algorithm to address multi-cut partitions, which will allow segmentation into multiple semantic categories beyond binary distinctions. Discussions on such extensions are included in the appendix. Additionally, there is an opportunity to explore more intricate graph structures that capture a broader spectrum of spatial information, enriching the segmentation process with greater detail and accuracy. Concurrently, an exploration into the capabilities of universal gate-based quantum computers for segmentation tasks is underway, focusing on the challenges posed by large-scale real-world images and leveraging the diverse technologies of existing hardware. These initiatives are driven by the goal of unlocking new capabilities in quantum computing to fully exploit its potential for tasks in computer vision.

APPENDIX

In this appendix, we discuss several possible extensions to the *Q-Seg* algorithm that could enhance its applicability and performance in various segmentation tasks. These formulations are exploratory and intended to provide a foundation for future research rather than final, fully-tested algorithms.

Extension to Multiple Classes

For multi-class segmentation involving *p* distinct classes, a potential extension of our approach is to create *p* individual binary segmentation tasks. Each task would segregate one particular class from all others, effectively converting a multi-class problem into multiple binary segmentation problems. In this proposed method, we define *p* different sets of edge weights, $w_1, w_2, \ldots, w_p$, where each set ($k = 1, 2, \ldots, p$) is specifically calibrated to highlight the boundaries of the *k*-th class. The objective is to compute *p* min-cut problems, each yielding a binary mask that segments the *k*-th class from the rest.

The QUBO formulation for each of these binary segmentation tasks can be represented as:

$$x_k^* = \arg\min_x \sum_{1 \leq i < j \leq n} x_{v_i}(1 - x_{v_j}) w_k(v_i, v_j)$$

where $x = x_{v_1}, x_{v_2}, \ldots, x_{v_n}$ and $x_{v_i} \in \{0, 1\}$ for the *k*-th task. Here, x_k^* represents the optimal binary segmentation for the *k*-th class. The weight function $w_k(v_i, v_j)$ encodes the edge weights for segmenting the *k*-th class, capturing the distinction between the pixels belonging to the class and those that do not.

By repeating this process for each class, we obtain *p* binary masks, each isolating one of the *p* classes. These masks can then be combined to construct a final multi-class segmentation of the image, demonstrating

the potential flexibility and scalability of *Q-Seg*. However, it is important to note that this multi-class extension is currently theoretical and has not been evaluated experimentally. Future work will focus on testing and validating this approach to assess its practical efficacy and performance.

Learning the Edge Weights

An alternative approach for defining the edge weight metric—a critical aspect of *Q-Seg* (or any other graph-based segmentation method), particularly in scenarios with sparse training data. By reverse-engineering the annotated segmentation masks available in such limited datasets, we can derive an empirical methodology for estimating the edge weight function. This strategy not only mitigates the issue of insufficient training data but also enhances the adaptability of *Q-Seg* to diverse applications by providing a tailored formulation for edge weight computation.

Problem Formulation: Let I be an image with pixels indexed by i. Let M be the corresponding binary segmentation mask, where $M_i = 1$ if pixel i is in the foreground and $M_i = 0$ if it's in the background. Let $p(i)$ be the set of neighboring pixels of pixel i.

Feature Extraction: For the evaluation of edge weights between each pair of neighboring pixels (i, j), with j being a member of the neighborhood $p(i)$, a multi-dimensional feature vector x_{ij} is meticulously constructed to encapsulate attributes like similarity/dissimilarity, gradient information, texture features, and statistical descriptors or color space transformations. This approach, by integrating a concise yet comprehensive set of information—brightness, contours, surface patterns, and application-specific criteria—enhances the determination of edge weights for graph-based image segmentation, aiming to accurately capture the complexity and diversity of images.

Target Weight Assignment (w^*): Assigning a target weight w_{ij}^* for each pair (i, j):

$$w_{ij}^* = \begin{cases} -1 & \text{if } M_i \neq M_j \quad \text{(boundary)} \\ +1 & \text{if } M_i = M_j \quad \text{(no boundary)} \end{cases} \quad (7)$$

Here, $w_{ij}^* = -1$ indicates a boundary (low weight for edges between different segments), and $w_{ij}^* = +1$ indicates no boundary (high weight for edges within the same segment). Bipolar labeling enhances learning efficiency and accuracy in models, especially neural networks with tanh activation. It simplifies edge weight interpretation, aiding in precise boundary and ensuring balanced weight distribution for improved boundary detection and region consistency. This method enhances

generalization and optimizes the min-cut algorithm by equally valuing boundary presence and absence.

Optimization Problem: We aim to learn a function $f(\mathbf{x}_{ij}; \theta)$ that maps the feature vector $\mathbf{x}_{ij}$ to an estimated weight w_{ij}, approximating the ideal weight w_{ij}^*. The function is parameterized by θ. The objective is to find parameters that minimize a loss function L over our labeled dataset, which quantifies the difference between the predicted weights w_{ij} and the target weights w_{ij}^*.

Let's say we have a dataset of N images $\{I^1, I^2, \dots, I^N\}$ and their corresponding binary segmentation masks $\{M^1, M^2, \dots, M^N\}$. For each image I_k and its corresponding mask M_k, we extract feature vectors x_{ij}^k and target weights w_{ij}^{*k} for each pair of neighboring pixels (i, j). The loss function $L(\theta)$ for the entire dataset is formulated as the average of the individual losses computed for each image:

$$L(\theta) = -\frac{1}{N}\sum_{k=1}^{N}\frac{1}{n_k}\sum_{(i,j)}\left[w_{ij}^{*k}\log f(\mathbf{x}_{ij}^k; \theta) \right.$$
$$\left. +(1 - w_{ij}^{*k})\log(1 - f(\mathbf{x}_{ij}^k; \theta)) \right] \quad (8)$$

where n_k is the number of pixel pairs in image I_k, w_{ij}^{*k} is the target weight for the pixel pair (i, j) in image I_k, determined by the segmentation mask M_k and, x_{ij}^k is the feature vector for the pixel pair (i, j) in image I^k.

Model Selection: The function f can be modeled using various machine learning algorithms. The choice depends on the dataset's complexity and the features.

Using Learned Weights in Min-Cut: After training, use f to assign weights in our graph for *Q-Seg*:

$$w_{ij} = f(\mathbf{x}_{ij}; \hat{\theta}) \quad (9)$$

where $\hat{\theta}$ are the optimized parameters.

These potential extensions require further testing and validation to determine their practical efficacy and performance.

Code Availability

All code to generate the data, figures, analyses, as well as, additional technical details on the experiments are publicly available at https://github.com/supreethmv/Q-Seg.

ACKNOWLEDGMENTS

REFERENCES

1. O. Ronneberger, P.Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, ser. LNCS, vol. 9351. Springer, 2015, pp. 234–241, (available on arXiv:1505.04597 [cs.CV]). [Online]. Available: http://lmb.informatik.uni-freiburg.de/Publications/2015/RFB15a

2. Y. Wang, J. Cheng, Y. Chen, S. Shao, L. Zhu, Z. Wu, T. Liu, and H. Zhu, "FVP: Fourier visual prompting for source-free unsupervised domain adaptation of medical image segmentation," *IEEE Transactions on Medical Imaging*, pp. 1–1, 2023. [Online]. Available: https://doi.org/10.1109%2Ftmi.2023.3306105

3. B. Solanki and M. Ingle, "Performance evaluation of thresholding techniques on modi script," in *2018 International Conference on Advanced Computation and Telecommunication (ICACAT)*, 2018, pp. 1–6.

4. D. Comaniciu and P. Meer, "Robust analysis of feature spaces: color image segmentation," in *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 1997, pp. 750–755.

5. I. Kotaridis and M. Lazaridou, "Remote sensing image segmentation advances: A meta-analysis," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 173, pp. 309–322, 2021.

6. S. Gould, T. Gao, and D. Koller, "Region-based segmentation and object detection," in *Advances in Neural Information Processing Systems*, Y. Bengio, D. Schuurmans, J. Lafferty, C. Williams, and A. Culotta, Eds., vol. 22. Curran Associates, Inc., 2009. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2009/file/a7aeed74714116f3b292a982238f83d2-Paper.pdf

7. F. Yi and I. Moon, "Image segmentation: A survey of graph-cut methods," in *2012 international conference on systems and informatics (ICSAI2012)*. IEEE, 2012, pp. 1936–1941.

8. S. Danda, A. Challa, B. S. Daya Sagar, and L. Najman, "Revisiting the isoperimetric graph partitioning problem," *IEEE Access*, vol. 7, pp. 50 636–50 649, 2019.

9. H. T. Larasati, T.-T.-H. Le, and H. Kim, "Trends of quantum computing applications to computer vision," in *2022 International Conference on Platform Technology and Service (PlatCon)*, 2022, pp. 7–12.

10. L. Tse, P. Mountney, P. Klein, and S. Severini, "Graph cut segmentation methods revisited with a quantum algorithm," *arXiv preprint arXiv:1812.03050*, 2018.

11. T. Presles, C. Enderli, G. Burel, and E. H. Baghious, "Synthetic aperture radar image segmentation with quantum annealing," *IET Radar, Sonar & Navigation*, vol. n/a, no. n/a, p. 1–13, 2024. [Online]. Available: https://ietresearch.onlinelibrary.wiley.com/doi/abs/10.1049/rsn2.12523

12. M. Ayodele, "Penalty weights in qubo formulations: Permutation problems," in *Evolutionary Computation in Combinatorial Optimization*, L. Pérez Cáceres and S. Verel, Eds. Cham: Springer International Publishing, 2022, pp. 159–174.

13. F. Glover, G. Kochenberger, R. Hennig, and Y. Du, "Quantum bridge analytics i: a tutorial on formulating and using qubo models," *Annals of Operations Research*, vol. 314, no. 1, pp. 141–183, Jul 2022. [Online]. Available: https://doi.org/10.1007/s10479-022-04634-2

14. A. Ng, M. Jordan, and Y. Weiss, "On spectral clustering: Analysis and an algorithm," in *Advances in Neural Information Processing Systems*, T. Dietterich, S. Becker, and Z. Ghahramani, Eds., vol. 14. MIT Press, 2001. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2001/file/801272ee79cfde7fa5960571fee36b9b-Paper.pdf

15. F. Hadlock, "Finding a maximum cut of a planar graph in polynomial time," *SIAM Journal on Computing*, vol. 4, no. 3, pp. 221–225, 1975. [Online]. Available: https://doi.org/10.1137/0204019

16. B. Monien and I. Sudborough, "Min cut is np-complete for edge weighted trees," *Theoretical Computer Science*, vol. 58, no. 1, pp. 209–229, 1988. [Online]. Available: https://www.sciencedirect.com/science/article/pii/030439758890028X

17. S. M. Venkatesh, A. Macaluso, and M. Klusch, "Gcs-q: Quantum graph coalition structure generation," in *Computational Science – ICCS 2023*, J. Mikyška, C. de Mulatier, M. Paszynski, V. V. Krzhizhanovskaya, J. J. Dongarra, and P. M. Sloot, Eds. Cham: Springer Nature Switzerland, 2023, pp. 138–152.

18. ——, "Bilp-q: Quantum coalition structure generation," in *Proceedings of the 19th ACM International Conference on Computing Frontiers*, ser. CF '22. New York, NY, USA: Association for Computing Machinery, 2022, p. 189–192. [Online]. Available: https://doi.org/10.1145/3528416.3530235

19. K. Iselborn, M. Stricker, T. Miyamoto, M. Nuske, and A. Dengel, "On the importance of feature representation for flood mapping using classical machine learning approaches," *arXiv preprint arXiv:2303.00691*, 2023.

20. A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo, P. Dollár, and R. Girshick, "Segment anything," *arXiv:2304.02643*, 2023.

Supreeth Mysore Venkatesh is pursuing a Ph.D. at the University of Kaiserslautern, Germany. His research focuses on Quantum Algorithms, Machine Learning, and Complexity Theory. Mysore Venkatesh received his Master's in Mathematics and Computer Science from Saarland University, Germany, in 2022. Contact him at supreeth.mysore@rptu.de.

Antonio Macaluso is a Senior Researcher at the German Research Center for Artificial Intelligence (DFKI), Saarbruecken, Germany, 66123 since 2022. His research interests span quantum algorithms in AI, including supervised and reinforcement learning, and multi-agent systems. Macaluso received his Ph.D. in Computer Science and Engineering from the University of Bologna, Italy in 2021. Contact him at antonio.macaluso@dfki.de.

Marlon Nuske received his Master's and PhD degrees in Physics from the University of Hamburg in 2015 and 2020, respectively. He is now working at the German Research Center for Artificial Intelligence (DFKI) in Kaiserslautern as a Senior Researcher leading the Earth and Space Applications team since 2021. His research interests lie in machine learning applications in earth observation, data fusion, hybrid modeling techniques, and physics-aware machine learning. Contact him at marlon.nuske@dfki.de.

Matthias Klusch is a Principal Researcher and Research Fellow at the German Research Center for Artificial Intelligence (DFKI) in Saarbruecken, Germany. His research interests include hybrid neuro-symbolic AI, semantic technologies, and quantum AI. Klusch received his habilitation in computer science from Saarland university. Contact him at www.dfki.de/ klusch.

Andreas Dengel is a Scientific Director at the German Research Center for Artificial Intelligence (DFKI GmbH), Kaiserslautern, Germany. His research interests include pattern recognition, document understanding, and information retrieval. He received his Ph.D. in computer science from the University of Stuttgart. He is a Fellow of the International Association for Pattern Recognition (IAPR). Contact him at andreas.dengel@dfki.de