

Harnessing DEN models for quantum computing tasks on neutral atom QPUs

Original

Harnessing DEN models for quantum computing tasks on neutral atom QPUs / Vercellino, C., Vitali, G., Viviani, P., Scionti, A., Terzo, O., Montrucchio, B.. - ELETTRONICO. - (2024), pp. 217-222. (2024 IEEE International Conference on Quantum Computing and Engineering (QCE) Montreal (CA) September 15–20, 2024) [10.1109/QCE60285.2024.10281].

Availability:

This version is available at: 11583/2992948 since: 2024-09-30T18:01:42Z

Publisher:

IEEE Computer Society

Published

DOI:10.1109/QCE60285.2024.10281

Terms of use:

This article is made available under terms and conditions as specified in the corresponding bibliographic description in the repository

Publisher copyright

IEEE postprint/Author's Accepted Manuscript

©2024 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collecting works, for resale or lists, or reuse of any copyrighted component of this work in other works.

(Article begins on next page)

Harnessing DEN models for quantum computing tasks on neutral atom QPUs

Chiara Vercellino^{*†‡}, Giacomo Vitali^{*†}, Paolo Viviani^{*}, Alberto Scionti^{*}, Olivier Terzo^{*}, Bartolomeo Montrucchio[†]

^{*}Fondazione LINKS, Torino, Italy

[†]DAUIN, Politecnico di Torino, Torino, Italy

[‡]chiara.vercellino@linksfoundation.com

Abstract—We present our work on effectively representing unit-disk graphs on the registers of neutral atom quantum machines. Specifically, we aimed to embed graphs corresponding to proteins and cellular antenna networks into unit-disk graphs, ensuring compatibility with the registers of two real QPUs: Orion Alpha by PASQAL and Aquila by QuEra. To address machine-specific constraints, we made adjustments and integrated *Distance Encoder Networks (DEN)* from our previous work. Despite these challenges, we successfully embedded up to 76% of protein-representing graphs for a quantum machine learning classification task on the Aquila QPU, and all subgraphs derived from 90 antenna geographical positions in Turin, Italy, on the Orion Alpha QPU. In the latter case, the graphs represented instances of the graph coloring problem, which we tackled using the hybrid quantum-classical algorithm *BBQ-mIS*. These promising results underscore the effectiveness and versatility of our embedding approach for representing unit-disk graphs on neutral atom quantum computers across diverse applications.

Index Terms—embedding, quantum optimization, quantum machine learning, neutral atoms, QPU

I. INTRODUCTION

Neutral atom quantum machines are a cutting-edge technology that harnesses the principles of quantum mechanics to perform computational tasks. At their core, these machines utilize individual neutral Rubidium (Rb) atoms as qubits.

One fascinating application of neutral atom quantum machines is their ability to represent natively through their machine Hamiltonian unit-disk graphs (UDG). Unit-disk graphs are a type of geometric graph where vertices represent points in a plane, and edges connect vertices if the corresponding points are within a unit distance of each other [1].

Neutral atom quantum machines offer a unique advantage in representing unit-disk graphs due to their inherent properties. By positioning the atoms in the quantum machine 2D register and exciting them from the ground state $|g\rangle$ to the excited Rydberg state $|r\rangle$ by laser pulses, the quantum system evolves according to the Hamiltonian \mathcal{H} . Specifically, if we consider a system of n qubits, laser settings characterized by Ω Rabi frequency and δ detuning and the position in the register of the i -th atom is p_i , then the machine Hamiltonian is expressed as in (1). The coefficient C_6 is related to the energy level of the Rb atoms used to represent the $|r\rangle$ state ($C_6 = 862690 \times 2\pi$

MHz μm^6 for $|r\rangle = |70S_{1/2}\rangle$ of the ^{87}Rb atoms), \hbar is the reduced Plank constant [2].

$$\mathcal{H} = \frac{\Omega\hbar}{2} \sum_{i=1}^n (|g_i\rangle\langle r_i| + |r_i\rangle\langle g_i|) - \delta\hbar \sum_{i=1}^n \hat{n}_i + \sum_{i<j} V_{ij} \quad (1)$$

$$V_{ij} = C_6 \frac{\hat{n}_i \hat{n}_j}{|p_i - p_j|^6} \quad (2)$$

$$\hat{n}_i = |r_i\rangle\langle r_i| \quad (3)$$

As highlighted by the V_{ij} term (2), interactions between pairs of qubits decrease with distance. In fact, by specifying values of Ω and δ , we can characterize the Rydberg radius $r_b = \sqrt[6]{\frac{C_6}{\hbar\sqrt{\Omega^2 + \delta^2}}}$. According to r_b , the qubits can incur into the blockade effect: r_b acts as a threshold distance, atoms closer than r_b cannot be simultaneously in the state $|r\rangle$, thus positioning atoms and setting the laser pulses parameters, it is possible to represent UDGs, whose vertices have the coordinates of the corresponding qubits in the register and the edges are present whenever qubits are closer than r_b [3], [4]. By encoding the graph's vertices onto individual qubits, neutral atoms machines can efficiently explore and process the graph's connectivity and properties to solve problems of interest, such as combinatorial graph optimization or graphs classification.

In our previous research [5], [6], we introduced a neural-network-based heuristic designed to embed Unit Disk Graphs (UDGs) into a prototype neutral atoms platform. We identified certain necessary conditions for graphs to possess feasible unit-disk embeddings, operating under the assumption that atoms could be freely positioned anywhere within the register, *i.e.*, free-space assumption.

However, upon transitioning to experimentation with two real Quantum Processing Units (QPUs), specifically Orion Alpha by PASQAL and Aquila by QuEra, we encountered additional hardware requirements. Our investigations spanned two distinct applications: *i)* On Orion Alpha, we implemented the *BBQ-mIS* algorithm outlined in [7]. This algorithm represents a hybrid quantum-classical approach aimed at solving graph coloring problems. *ii)* On Aquila, we conducted tests involving a quantum machine learning algorithm inspired by [8], which aims to classify proteins as either enzymes or non-enzymes.

For our graph coloring application, our dataset comprises 90 antennas located in the city of Turin, extracted from a publicly available website¹. The graph coloring problems are derived from the optimization of Principal Cell Identifiers (PCIs), where each antenna must be efficiently assigned a PCI to minimize conflicts. Specifically, antennas within overlapping coverage areas should not share the same PCI, and the total number of PCIs must be minimized. To address this application, we adapted the solutions obtained using the DEN model [6] to map the coordinates in free-space to a predefined set of fixed positions. These positions were determined by a triangular lattice with a side length of $5\mu m$.

For graph classification, we embedded samples from the PROTEINS dataset [9], [10] into the Aquila register. To ensure the feasibility of these embeddings, we introduced an additional constraint. While the DEN model accounted for the unit-disk constraint and determined the minimum/maximum distances among atoms, our experimental setup required the inclusion of a row-spacing constraint. During the preparation of the register, qubits were mandated to either belong to the same row or possess a y-axis distance greater than or equal to $4\mu m$. This constraint was crucial for accommodating the physical layout of the register and ensuring the viability of our experimental runs.

II. METHODOLOGY

Because of the unique characteristics of the Aquila and Orion Alpha machines, we opted to adapt and integrate the DEN model’s embedding method in two distinct manners. Employing a nearest neighbor approach, we transformed the free-space solutions into positions arranged according to a triangular-based lattice layout (Orion Alpha’s QPU). For embedding the graphs derived from the PROTEINS dataset for the classification task, we introduced a new component to the *Embedding Loss Function (ELF)* [6]. This addition was crucial for accurately representing the row-spacing constraint, ensuring that our embeddings aligned with the physical constraints of the quantum hardware (Aquila’s QPU).

A. UDG embedding on Orion Alpha: DEN models and nearest neighbor techniques

To embed the graph coloring samples onto Orion Alpha hardware, we employed a novel approach involving the training of a DEN model on each sample. Unlike traditional predictive models, the DEN operates as an optimization solver [5], [6]. For each new instance of the constrained unit-disk graph (CUDG) problem, the DEN learns to generate a feasible embedding by leveraging an initial unfeasible yet easily retrievable solution.

This learning process leverages the *Embedding Loss Function (ELF)*, which imposes penalties by assigning positive values within the loss function to be minimized. These penalties are applied to address constraint violations, ensuring that the DEN model adjusts and learns to generate embeddings that

conform to specific constraints inherent in the Constrained Unit Disk Graph (CUDG) problem.

- Atom pairs must maintain a distance $\geq 5\mu m$ but $\leq 40\mu m$ to adhere to hardware register constraints.
- For the unit-disk constraint, atom pairs corresponding to adjacent vertices must be separated by a distance $\leq 10.26\mu m$, while atom pairs corresponding to non-adjacent vertices must have a distance $> 10.26\mu m$.

As anticipated, the actual Quantum Processing Unit (QPU) we intended to utilize mandates that atom positions, representing the vertices of the UDGs, be selected from a predefined set of fixed positions arranged in a triangular lattice layout. This requirement aims to expedite algorithm execution times, particularly due to the calibration procedure. The available positions in the lattice correspond to the coordinates of optical traps within the register, with 61 traps arranged in a triangular layout with a side length of $5\mu m$.

Given that the DEN model was not originally designed to determine atom positions in discrete space, and integrating the corresponding integrality constraint into the *ELF* proved infeasible due to non-differentiable points in the loss function, our initial approach was to solve the Mixed Integer Program (MIP) using state-of-the-art solvers such as *Gurobi* [11], *Ipopt* [12], and *GLPK*. However, this approach led to quadratic or polynomially increasing linear constraints, rendering it impractical to retrieve feasible embedding solutions within a time limit of 5 hours, even for small graph instances comprising 7-10 vertices. Consequently, we sought an alternative solution to map the DEN solutions to the positions of the optical traps.

Our final approach, outlined in Algorithm 1, begins with a DEN solution and proceeds to sort the vertices in descending order of degree. Subsequently, it identifies the closest trap position for each vertex. While this approach does not guarantee the preservation of a unit-disk representation of the graph, it consistently yields hardware-feasible embeddings. The decision to prioritize vertices with higher degrees in the sorting process is motivated by the objective of positioning vertices with more complex connectivity, *i.e.*, those with a greater number of adjacent vertices, at earlier stages of the embedding process.

Algorithm 1 Embedding algorithm on Orion Alpha

```

P ← DEN solution for  $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ 
T ← trap coordinates in the triangular-based lattice
 $\mathcal{V}_s \leftarrow \text{sort}(\mathcal{V})$            ▷ sort  $\mathcal{V}$  by decreasing degree
for  $v \in \mathcal{V}_s$  do
     $p \leftarrow P(v)$            ▷ (x,y) coordinates of  $v$  in  $P$ 
     $p' \leftarrow$  nearest neighbor of  $p$  in  $T$ 
     $T \leftarrow T \setminus p'$ 
     $P(v) \leftarrow p'$ 
end for

```

¹<https://opencellid.org>

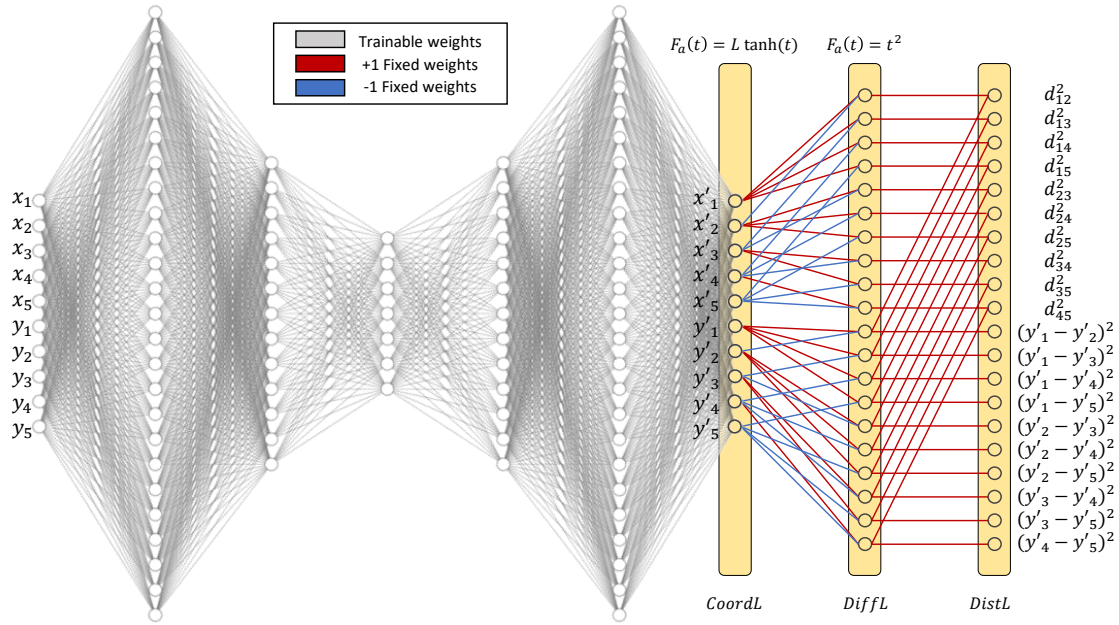


Fig. 1. Modified *DEN* model architecture for feasible embeddings on Aquila QPU - an example of embedding a 5-vertex graph.

B. UDG embedding on Aquila: ELF with row spacing constraints

For UDG embeddings on the Aquila register, we were able to maintain our reliance on free-space assumptions. However, compared to the prototype considered in the definition of the *DEN* model, we encountered two significant differences [2].

Firstly, the Aquila register does not possess a circular shape; instead, it is rectangular, measuring $75\mu m \times 76\mu m$. This rectangular shape proved advantageous from the perspective of *ELF* design, as it simplified the modeling of atom placement within the register. By utilizing the activation function in the *CoordL*, we could easily represent atom belonging to the register. Setting the parameter $L = 75/2$ enabled us to generate (x, y) coordinates $\in (-75/2, +75/2)^2$. Consequently, by applying a simple translation to the *DEN* solutions along the x and y axis, we were able to obtain embeddings that could be accommodated within the register.

A critical hardware constraint arises from the register loading procedure, significantly impacting the embedding process. To expedite the loading of atoms into the QPU, they are organized into rows within the register. This necessitates that rows be spaced at least $4\mu m$ apart. In order to accommodate this row-spacing constraint, we made modifications to the *DEN* architecture, as illustrated in Fig. 1, which depicts the model for a 5-vertex embedding.

Similar to the previous version, the input consists of an infeasible embedding solution represented by the vertex coordinates $x_1, \dots, x_5, y_1, \dots, y_5$. These coordinates are passed through an autoencoder comprising fully connected and trainable layers (depicted in gray). At the final layer of the autoencoder, we apply an activation function defined as $F_a(t) = L \cdot \tanh(t)$, yielding new coordinates $x'_1, \dots, x'_5, y'_1, \dots, y'_5$

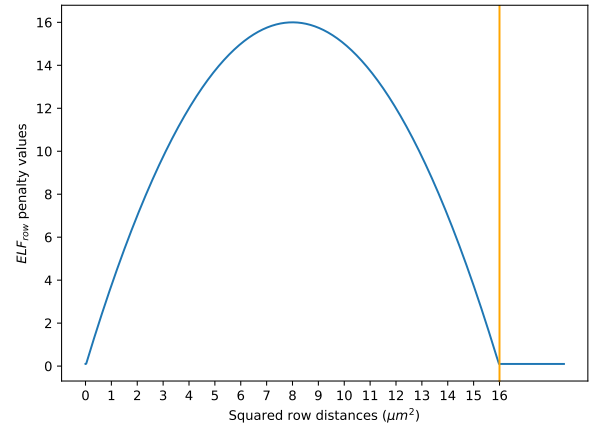


Fig. 2. ELF_{row} component for incorporating the row-spacing constraint. ELF_{row} equals 0 in cases where two vertices either share the same row or are separated by a row-distance greater than $4\mu m$ (squared distance $\geq 16\mu m^2$).

that align with the rectangular register. Subsequent layers possess fixed weights, meaning they are not updated by the *AdamW* optimizer responsible for optimizing the autoencoder weights.

The fixed-weight layers are sparsely populated with values of ± 1 . The *DiffL* layer computes the pairwise differences along both the x and y axes, namely $x'_1 - x'_2, x'_1 - x'_3, \dots, x'_4 - x'_5, y'_1 - y'_2, y'_1 - y'_3, \dots, y'_4 - y'_5$. On the *DiffL* layer, we apply the square activation function $F_a(t) = t^2$. The most significant departure from the *DEN* version presented in [6] is evident in the output layer, *DistL*. This layer encompasses both the squared pair differences between each vertex pair i, j , $d_{ij}^2 = (x'_i - x'_j)^2 + (y'_i - y'_j)^2$ and the squared row distance

$(y'_i - y'_j)^2$. Considering a graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$, with \mathcal{V} as the set of vertices, \mathcal{E} as the set of undirected edges, and $n = |\mathcal{V}|$, the *DistL* layer consists of $2 \times \binom{n}{2}$ elements. The first $\binom{n}{2}$ output elements contribute to the computation of the ELF_{min} and ELF_{max} components of the loss function, as described in [6]. When the sum of ELF_{min} and ELF_{max} is 0, it indicates that the coordinates $(x'_1, y'_1), \dots, (x'_n, y'_n)$ corresponds to a unit-disk feasible embedding. In this configuration, adjacent vertices are positioned closer than $10.26\mu m$ and further apart than $4\mu m$, while non-adjacent vertices are separated by distances exceeding $10.26\mu m$. Furthermore, they inherently adhere to the register dimensions due to the definition of F_a in the *CoordL*. To incorporate the row-spacing constraint into the loss function, we introduced a new component, ELF_{row} , which operates on the last $\binom{n}{2}$ elements in the *DistL* layer. ELF_{row} combines a parabolic penalty, peaking at a row distance of $2\sqrt{2}\mu m$, with a threshold function to prevent negative contributions from row distances exceeding $4\mu m$ (refer to Fig. 2).

When the sum of ELF_{min} , ELF_{max} , and ELF_{row} is 0, we achieve a feasible embedding for the Aquila platform.

III. RESULTS AND DISCUSSION

To evaluate the effectiveness of these enhanced versions of the DEN model, we conducted experiments involving graph embeddings for two distinct applications. The first application pertained to a combinatorial optimization task, specifically the resolution of the graph coloring problem. The second application focused on a quantum machine learning scenario for graph classification.

In discussing the results of these embeddings, we introduce the following notation: the graph to be embedded is denoted as $\mathcal{G}(\mathcal{V}, \mathcal{E})$, with n vertices. We define $D_{adj} := \max_{(i,j) \in \mathcal{E}} d_{ij}$ as the maximum distance between adjacent vertices determined by the DEN model, and $d_{-adj} := \min_{(i,j) \notin \mathcal{E}} d_{ij}$ as the minimum distance between non-adjacent vertices. Feasible embeddings are identified by satisfying $D_{adj} \leq 10.26\mu m$ and $D_{adj} < d_{-adj}$. The larger the difference $d_{-adj} - D_{adj}$, the higher the quality of the embedding, enabling better discrimination between adjacent and non-adjacent vertices in the presence of noise. Additionally, Δ_{max} represents the maximum degree in \mathcal{G} , while $|K_{max}|$ indicates the size of the maximum estimated fully-connected subgraph in \mathcal{G}^2 .

As discussed in [5], two necessary conditions to embed a unit-disk graph under the free-space assumption, considering a minimum distance of $4\mu m$ among vertices and a maximum distance $D_{adj} \leq 10.26\mu m$, are $\Delta_{max} \leq 18$ and $|K_{max}| \leq 7$.

A. Graph coloring

For the graph coloring application, the original dataset represents a cellular antenna network comprising 90 antennas. These networks can naturally be represented as unit-disk graphs, as their conflicts are dependent on distances (in our case, the conflict distance is $140m$). However, obtaining

an Orion Alpha-feasible representation is not achievable by simple rescaling due to the minimum distance requirement among atoms and the fixed qubit positions.

Therefore, we utilize a rescaled embedding as an infeasible solution, serving as input for the *DEN* model. Subsequently, we optimize the trainable weights to attain a feasible solution under the free-space assumption, with $L = 41/2$. The maximum distance among Orion Alpha register traps in the lattice is $40\mu m$, and the minimum distance among atoms is $5\mu m$. We then apply Algorithm 1 to the *DEN* solutions to map coordinates from free-space to the triangular lattice.

Given that the total number of antennas, and thus the number of vertices to be mapped to the QPU register, is 90, exceeding the available 61 atoms, we identify connected components within the antenna network. Each connected component is embedded separately. Upon subdividing the original network into connected components, we obtain:

- 9 isolated vertexes
- 6 K_2 , 4 K_3 , 1 K_4 , 2 K_5 complete graphs, whose unit-disk embeddings are easily identifiable and for which the graph coloring problem has a trivial singleton solution
- 2 7-vertex graphs ($7G_{3cc}$ and $7G_{13cc}$), 1 10-vertex graph ($10G_{4cc}$), and 1 19-vertex graph ($19G_{1cc}$)

Isolated vertices and complete graphs have trivial graph coloring solutions and embeddings. Therefore, our focus was on the embeddings of graphs $7G_{3cc}$, $7G_{13cc}$, $10G_{4cc}$ and $19G_{1cc}$. Table I provides a summary of the results and characteristics of these graphs. D_{adj}^l and d_{-adj}^l represent the values of D_{adj} and d_{-adj} computed on the lattice coordinates obtained with Algorithm 1. Under the free-space assumption, the DEN model successfully found feasible embeddings for all four graphs. However, the remapping algorithm generally deteriorated the quality of the embedding solution, as $d_{-adj}^l - D_{adj}^l$ was consistently lower than $d_{-adj} - D_{adj}$. In some cases, this difference was 0, suggesting a need for improvement in the remapping process. Nevertheless, the *BBQ-mIS* algorithm [7], used to solve the corresponding graph coloring problem, proved robust enough to handle these non-ideal unit-disk configurations and find optimal colorings regardless. This resilience can be attributed to the small number of edges that are not properly represented. Figure 3 illustrates the remapped embedding of graph $19G_{1cc}$, where we observe that considering a unit-disk radius $r_b = 10.0\mu m$ would result in only one additional edge (colored in orange) connecting vertices 13 and 22. The issue of $d_{-adj}^l - D_{adj}^l = 0$ can be resolved by maintaining the left part of the graph's vertex positions as they are and applying the following vertex \rightarrow trap remappings: $22 \rightarrow 45$, $31 \rightarrow 49$, $37 \rightarrow 53$, $44 \rightarrow 59$, $40 \rightarrow 58$ and $49 \rightarrow 56$ (previously 44 vertex trap).

B. Proteins classification

In the case of the protein classification task, we collected graph samples from the PROTEINS dataset [9], [10], and applied the methodology described in Sec. II-B. Given that the maximum number of atoms, *i.e.*, qubits, in Aquila's QPU is 256, we specifically selected graph samples with $n \leq 256$.

²We used *networkx* heuristic to estimate $|K_{max}|$

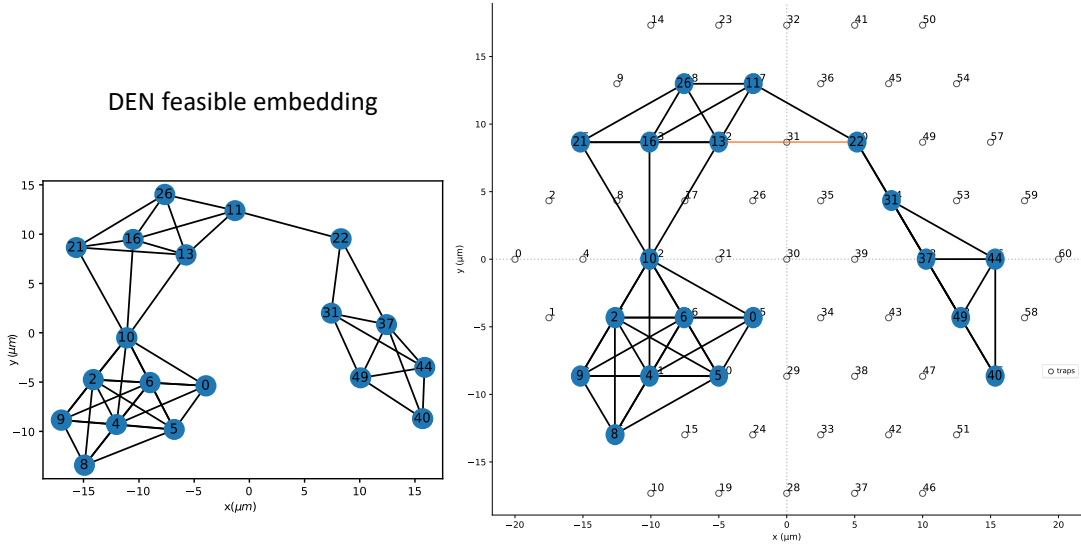


Fig. 3. Mapping of the feasible embedding for sample 19G_1cc to Orion Alpha register.

TABLE I
ANTENNAS DATASET EMBEDDINGS. ALL DISTANCES ARE IN μm .

Graph	D_{adj}	d_{-adj}	D_{adj}^l	d_{-adj}^l	Δ_{max}	$ K_{max} $
7G_3cc	10.2	16.3	10.0	13.2	5	4
7G_13cc	10.2	12.2	10.0	10.0	5	4
10G_4cc	10.1	15.7	10.0	10.0	4	4
19G_1cc	10.2	13.3	10.0	10.0	9	6

Furthermore, as our graph classification application encompassed both emulation of the quantum system on CPU and simulation on QPU, we divided the overall dataset into three subsets:

- PROTEINS_12 includes all the graph samples in the PROTEINS dataset with $n \leq 12$
- PROTEINS_16 includes all the graph samples in the PROTEINS dataset with $n \leq 16$
- PROTEINS_256 includes all the graph samples in the PROTEINS dataset with $n \leq 256$

We utilized the embedded samples of graphs in PROTEINS_12 and PROTEINS_16 for the CPU emulations. The outputs from these emulations were then employed to optimize the laser pulses and identify the best classifier hyperparameters. For both of these datasets, we obtained encouraging results. In the PROTEINS_12 dataset, we were able to embed 158 out of the total 207 graph samples (approximately 76% success rate). However, this success rate decreased as we increased the value of n . When considering the samples from the PROTEINS_16 dataset as well, we could obtain Aquila-feasible UDG embeddings for 210 graphs out of 307 samples (approximately 68% success rate). More detailed results are presented in Fig. 4, which illustrates the percentage of Aquila-feasible embeddings found by the DEN model across different

values of n . It is noteworthy that as n increases, finding feasible UDG embeddings becomes more challenging, leading to a lower success rate. A similar trend is evident in Fig. 5, which depicts the distributions of the $d_{-adj} - D_{adj}$ gap. As the dimensionality of the CUDG problem increases, it becomes increasingly difficult to find embeddings that effectively differentiate between adjacent and non-adjacent vertex pairs.

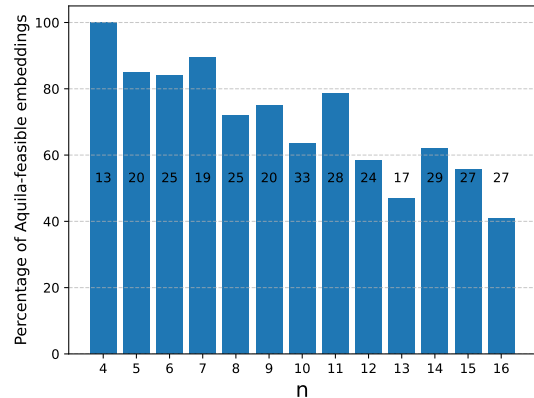


Fig. 4. Percentage of graph samples for which the DEN model found an Aquila-feasible embedding, grouped by the number of vertices n . The numbers reported in each bin represent the total number of graph samples for each value of n .

When we transitioned to the PROTEINS_256 dataset, obtaining feasible embeddings became even more challenging. We increased the total number of feasible embeddings to 279, which accounts for 34% of the overall dataset, this increment in dataset size proved beneficial for the classification task. However, the decreased success rate in embedding feasibility

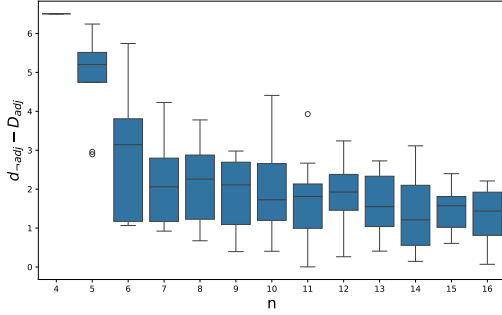


Fig. 5. Gap distributions of $d_{-adj} - D_{adj}$ grouped by the number of vertices n , for the graph-embedded samples in the PROTEINS_12 and PROTEINS_16 datasets.

highlights the limitation of the current approach, particularly when targeting graphs with n very close to the maximum number of qubits in the QPU.

This outcome is expected, given that positioning 256 vertices within a rectangle measuring $75\mu\text{m} \times 76\mu\text{m}$, while adhering to the minimum distance and row-spacing constraints, presents significant difficulties in achieving a feasible exact representation of a general graph.

To illustrate the complexity of the UDG embedding problem, we present in Fig. 6 the Aquila-feasible embedding of graph sample 115 in the PROTEINS_256 dataset. This graph comprises 37 vertices, with $\Delta_{max} = 7$ and $|K_{max}| = 4$. We achieved feasibility with values $D_{adj} = 10.0\mu\text{m}$ and $d_{-adj} = 10.4\mu\text{m}$.

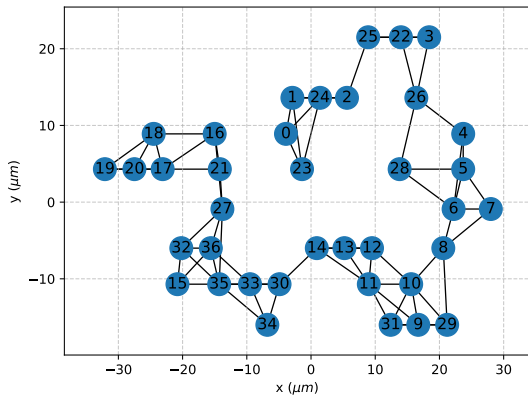


Fig. 6. DEN feasible embedding of sample 115 in the PROTEINS_256 dataset.

It is noteworthy that, even though the success rate for embedding on larger graphs might seem low, the success rate of state-of-the-art optimization solvers was close to 0% even on smaller graphs, despite the programming model formulation, and even allowing for very long computational times to find feasible solutions (5-10 hours). This proves the effectiveness of our heuristic, which usually requires less than 5 minutes to

provide a solution.

IV. CONCLUSION

In conclusion, this paper thoroughly investigates the embedding of constrained unit-disk graphs (CUDGs) onto neutral atom QPUs, with a focus on the Aquila by QuEra and Orion Alpha by PASQAL platforms. Our experiments and analyses highlight the challenges and constraints in achieving feasible embeddings, particularly as graph complexity and hardware limitations increase.

We address the current lack of tools for neutral atom users by providing a methodology for mapping their problems onto these machines. Despite the difficulty of solving CUDGs, as indicated by the lower success rate of the *DEN* model embedding on larger graphs, our results demonstrate that even partial successes represent progress. Furthermore, we envision that better utilization of available qubits can be achieved with smaller graphs by replicating the same graph multiple times within the register, ensuring sufficient distance between replicates to prevent interaction amongst qubits belonging to different copies. This replication strategy reduces the number of measurements required for quantum tasks, making the process faster and more cost-effective.

REFERENCES

- [1] B. N. Clark, C. J. Colbourn, and D. S. Johnson, "Unit disk graphs," *Discrete mathematics*, vol. 86, no. 1-3, pp. 165–177, 1990.
- [2] J. Wurtz, A. Bylinskii, B. Braverman, J. Amato-Grill, S. H. Cantu, F. Huber, A. Lukin, F. Liu, P. Weinberg, J. Long *et al.*, "Aquila: Quera's 256-qubit neutral-atom quantum computer," *arXiv preprint arXiv:2306.11727*, 2023.
- [3] D. Ciampini, O. Morsch, and E. Arimondo, "Ultracold rubidium atoms excited to rydberg levels," *Journal of Atomic, Molecular, Condensed Matter and Nano Physics*, vol. 2, no. 3, pp. 161–167, 2015.
- [4] C. Picken, R. Legaie, K. McDonnell, and J. Pritchard, "Entanglement of neutral-atom qubits with long ground-rydberg coherence times," *Quantum Science and Technology*, vol. 4, no. 1, p. 015011, 2018.
- [5] C. Vercellino, P. Viviani, G. Vitali, A. Scionti, A. Scarabosio, O. Terzo, E. Giusto, and B. Montrucchio, "Neural-powered unit disk graph embedding: qubits connectivity for some qubo problems," in *2022 IEEE International Conference on Quantum Computing and Engineering (QCE)*. IEEE, 2022, pp. 186–196.
- [6] C. Vercellino, G. Vitali, P. Viviani, A. Scionti, A. Scarabosio, O. Terzo, E. Giusto, and B. Montrucchio, "Neural optimization for quantum architectures: graph embedding problems with distance encoder networks," in *2023 IEEE 47th Annual Computers, Software, and Applications Conference (COMPSAC)*. IEEE, 2023, pp. 380–389.
- [7] C. Vercellino, G. Vitali, P. Viviani, E. Giusto, A. Scionti, A. Scarabosio, O. Terzo, and B. Montrucchio, "Bbq-mis: a parallel quantum algorithm for graph coloring problems," in *2023 IEEE International Conference on Quantum Computing and Engineering (QCE)*, vol. 2. IEEE, 2023, pp. 141–147.
- [8] L.-P. Henry, S. Thabet, C. Dalyac, and L. Henriot, "Quantum evolution kernel: Machine learning on graphs with programmable arrays of qubits," *Physical Review A*, vol. 104, no. 3, p. 032416, 2021.
- [9] K. M. Borgwardt, C. S. Ong, S. Schönauer, S. Vishwanathan, A. J. Smola, and H.-P. Kriegel, "Protein function prediction via graph kernels," *Bioinformatics*, vol. 21, no. suppl_1, pp. i47–i56, 2005.
- [10] P. D. Dobson and A. J. Doig, "Distinguishing enzyme structures from non-enzymes without alignments," *Journal of molecular biology*, vol. 330, no. 4, pp. 771–783, 2003.
- [11] B. Bixby, "The gurobi optimizer," *Transp. Re-search Part B*, vol. 41, no. 2, pp. 159–178, 2007.
- [12] A. Wächter and L. T. Biegler, "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming," *Mathematical programming*, vol. 106, no. 1, pp. 25–57, 2006.