Kernel k-Medoids as General Vector Quantization

1st Thore Gerlach
University of Bonn
Bonn, Germany
gerlach@iai.uni-bonn.de

2nd Sascha Mücke
TU Dortmund University
Dortmund, Germany
sascha.muecke@tu-dortmund.de

3rd Christian Bauckhage
Fraunhofer IAIS
Sankt Augustin, Germany
christian.bauckhage@iais.fraunhofer.de

***Abstract*—Vector Quantization (VQ) is a widely used technique in machine learning and data compression, valued for its simplicity and interpretability. Among hard VQ methods, k-medoids clustering and Kernel Density Estimation (KDE) approaches represent two prominent yet seemingly unrelated paradigms—one distance-based, the other rooted in probability density matching. In this paper, we investigate their connection through the lens of Quadratic Unconstrained Binary Optimization (QUBO). We compare a heuristic QUBO formulation for k-medoids, which balances centrality and diversity, with a principled QUBO derived from minimizing Maximum Mean Discrepancy in KDE-based VQ. Surprisingly, we show that the KDE-QUBO is a special case of the k-medoids-QUBO under mild assumptions on the kernel's feature map. This reveals a deeper structural relationship between these two approaches and provides new insight into the geometric interpretation of the weighting parameters used in QUBO formulations for VQ.**

***Index Terms*—Vector Quantization, QUBO, Kernel Density Estimation, k-Medoids, Quantum Computing**

I. Introduction

Vector Quantization (VQ) [1, 2] is a foundational method in machine learning and data compression, where the goal is to represent a large dataset using a significantly smaller subset of representative points known as prototypes. In hard VQ, the prototypes are selected directly from the dataset itself, ensuring that all representations are grounded in real, observed data. These methods are particularly valued for their intuitive problem formulation and interpretable model behavior [3]—traits that are increasingly important in domains such as healthcare [4], autonomous systems [5] or scientific discovery [6], where transparency and explainability are essential. Depending on the context, prototypes may serve different purposes: in unsupervised learning, they are used for data representation or clustering [7], while in supervised settings, they support classification or regression tasks [8].

Among the many approaches to VQ, two families of methods have emerged as particularly prominent: distance-based centroid selection and probability density estimation methods. The former includes algorithms like k-means [9] and k-medoids [10], where the goal is to identify a set of data points that minimize the average dissimilarity to the rest of the dataset. The latter group includes Kernel Density Estimation (KDE) approaches, where the similarity between the data distribution and the selected prototypes is optimized using tools like the kernel trick and information-theoretic concepts [11, 12]. While both k-medoids-based VQ (MED-VQ) and KDE-based VQ (KDE-VQ) can be interpreted as prototype selection problems, their conceptual and mathematical connections remain poorly understood. They appear, at first glance, to address distinct optimization goals—centrality versus distribution matching—and have evolved largely independently within the literature.

A significant challenge shared by hard VQ methods is their combinatorial nature. Finding optimal prototype sets is computationally intractable in general, with problems like k-medoids being NP-hard [13]. As the search space grows exponentially with the dataset size, exact solutions become impractical for real-world applications. This computational bottleneck has motivated the use of specialized hardware accelerators, particularly in solving Quadratic Unconstrained Binary Optimization (QUBO) [14] problems that arise in prototype selection, which are of the form

$$\min_{\boldsymbol{z}\in\{0,1\}^n} \boldsymbol{z}^\top \boldsymbol{Q}\boldsymbol{z}\,, \tag{1}$$

where $\boldsymbol{Q}$ is an $n \times n$ real matrix. Among these hardware accelerators, Ising machines [15] represent a notable direction. These devices are designed to solve optimization problems of the form given in Eq. (1) by finding the ground state of an Ising Hamiltonian, effectively translating optimization problems into physical energy minimization processes. Quantum Computing (QC) [16], in particular, offers a powerful model in this realm by exploiting superposition, entanglement, and quantum tunneling. One can differentiate between digital QC—coined Quantum Gate Computing (QGC)—which is used to implement sequential algorithmic structures via unitary operations and analog QC, such as Quantum Annealing (QA) [17], based on the principles of Adiabatic Quantum Computing (AQC) [18]. While fully scalable quantum computers remain in development, early devices—classified as Noisy Intermediate-Scale Quantum (NISQ) hardware [19]—are already accessible and have spurred growing interest in quantum optimization.

In this paper, we investigate the relationship between two QUBO formulations of hard VQ: one derived from k-medoids [20] clustering, and another from KDE divergence minimization [21]. The QUBO formulation for MED-VQ follows rather heuristic arguments where two competing objectives—the selection of central and mutually distant data points—are balanced via scalar weighting parameters. In opposition, a KDE-VQ QUBO is derived with a principled approach by minimizing the Maximum Mean Dis-

crepancy (MMD) between the full dataset and a candidate prototype subset. Both VQ methods are first formulated as Quadratic Binary Programming (QBP) [22] problems and then reformulated to QUBOs by incorporating linear constraints. Surprisingly, we find that the principled QUBO based on MMD minimization is a special case of the heuristic MED-VQ approach. This equivalence emerges under a mild assumption: that the kernel-induced feature map embeds data onto the unit sphere of the feature space—an assumption that holds for commonly used Radial Basis Function (RBF) or Laplacian kernels as well as quantum kernels [23]. Our contributions are threefold:

- We show that the KDE-VQ QUBO formulation is a special case of the MED-VQ QUBO, revealing a deeper structural relationship between distance-based and density-based VQ,
- By showing that the derived equivalence holds for normalized Mercer kernels (including RBF, Laplacian, and quantum kernels), the paper highlights a broader applicability of the findings. This generalization opens avenues for combining the AQC and QGC paradigms,
- We interpret the role of the scalar weighting parameter in the MED-VQ QUBO formulation as a scaling factor for the kernel-induced geometry, providing insights into its meaning for quantum-based optimization.

This paper is structured as follows: we first give some insights on related work in Sec. II, while Sec. III captures all necessities for obtaining QUBO formulations for MED-VQ and KDE-VQ. In Sec. V, we state our main theoretical insights and discuss them in Sec. V. Finally, we draw a conclusion in Sec. VI.

II. Related Work

A growing body of research explores how QC and quantum-inspired models can enhance or accelerate VQ. In [24], the authors integrate VQ into a hybrid framework that leverages quantum-inspired models to optimize prototype adaptation. In [25], the authors take a step further by directly formulating learning VQ (LVQ) for classification tasks on theoretical QGC architectures. The work in [26] explores similar ideas but focuses on how quantum principles, such as unitary operations and kernel-based feature embeddings, can be used to improve prototype-based classifiers.

Another promising direction involves leveraging AQC and related QUBO-based formulations to address the combinatorial nature of prototype selection in VQ. The foundation stone was layed in [20], where the authors come up with a QUBO formulation for the k-medoids problem, by balancing two competing objectives. This work is extended in [27], where a preferable choice of the balancing parameters is proposed. A different approach is taken in [28], where VQ is interpreted as a Set Cover problem, which is then reformulated into the QUBO framework. Finally, the authors of [21] utilize MMD for KDE to obtain a formulation suited for Hopfield networks, which are equivalent to QUBOs.

III. Background

For notational convenience, we indicate vectors by bold lowercase and matrices by bold capitalized letters. Further, we define $\mathbb{B} \coloneqq \{0,1\}^n$ and denote the vector consisting of only ones by $\mathbf{1}$, where its dimension is induced by the context. Moreover, the function $\mathrm{diag}[\cdot]$, which takes a vector as input, denotes a diagonal matrix with that vector as its diagonal.

Generally speaking, VQ is a technique used in signal processing and ML to compress a dataset into a finite set of representative points, known as prototypes. For $\mathcal{D} = \{\boldsymbol{x}^1, \dots, \boldsymbol{x}^n\} \subset \mathbb{X}$, it aims to identify $k \ll n$ prototypes $\mathcal{W} = \{\boldsymbol{w}^1, \dots, \boldsymbol{w}^k\} \subset \mathbb{X}$ that serve as a compressed representation of the dataset. These representative points form a codebook, and each input vector is mapped to its closest prototype, effectively quantizing the data.

It functions by partitioning a large set of points (vectors) into groups, with each group containing roughly the same number of points nearest to it. This makes VQ suitable for lossy data compression, pattern recognition and density estimation. Furthermore, it is very similar to clustering but describes a slightly different task.

A. k-Medoids

A simple VQ method is to use the cluster means of k-means clustering as prototypes. In hard VQ, the prototypes exactly correspond to the medoids of the clusters.

We focus on hard clustering, i.e., we want to find disjoint subsets $C_i \subset \mathcal{D}$ with $\bigcup_i C_i = \mathcal{D}$ such that points in C_i are similar and points from two different clusters C_i and C_j are dissimilar. Hence, one relies on suitable similarity measures such as Euclidean distance or cosine similarity. The k-medoids objective is to minimize the within cluster scatter

$$\min_{C_1,\dots,C_k} \sum_{i\in[k]} \sum_{\boldsymbol{x}\in C_i} D(\boldsymbol{x}, \boldsymbol{m}_i) \;, \tag{2}$$

where the medoid $\boldsymbol{m}$ is defined to be an element of C

$$\boldsymbol{m} = \arg\min_{\boldsymbol{y}\in C} \sum_{\boldsymbol{x}\in C} D(\boldsymbol{x}, \boldsymbol{y}) \;,$$

and $D : \mathbb{X} \times \mathbb{X} \to \mathbb{R}$ is a suitable distance measure. A notable characteristic of medoids is that they are determined solely by evaluating distances between given data points. Such distances can be precomputed and thus do not rely on numeric data, in opposition to means.

A local minimum of the objective in Eq. (2) can be obtained by using a slightly different versions of Lloyd's algorithm [9]. This method initializes k medoids and iteratively determines clusters by assigning data points to their closest medoid and updates these medoids according to that assignment. Finding a global optimum, however, is much harder, since k-medoids is NP-hard [13].

The authors of [20] propose a QUBO formulation for estimating Eq. (2). For this, *Fisher's analysis of variance* [29] is used, which states that the sum of the within cluster scatter and the between cluster scatter is constant. Minimizing the

within cluster scatter can be written as selecting the k most central data points

$$\begin{aligned}\min_{\mathcal{W}\subset\mathcal{D}} \sum_{\boldsymbol{x}\in\mathcal{W}}\sum_{\boldsymbol{y}\in\mathcal{D}} D(\boldsymbol{x},\boldsymbol{y})\\ \text{s.t. } |\mathcal{W}| = k\ ,\end{aligned}$$

while maximizing the between cluster scatter is similar to finding far apart data points

$$\begin{aligned}\max_{\mathcal{W}\subset\mathcal{D}} \sum_{\boldsymbol{x}\in\mathcal{W}}\sum_{\boldsymbol{y}\in\mathcal{W}} D(\boldsymbol{x},\boldsymbol{y})\\ \text{s.t. } |\mathcal{W}| = k\ ,\end{aligned}$$

where $\mathcal{W} = \{\boldsymbol{m}_1,\dots,\boldsymbol{m}_k\}$ denotes the set of cluster medoids. Putting both objectives together by weighting them with parameters $\alpha,\beta>0$, we get a single objective

$$\begin{aligned}\min_{\mathcal{W}\subset\mathcal{D}} \ -\alpha\sum_{\boldsymbol{x}\in\mathcal{W}}\sum_{\boldsymbol{y}\in\mathcal{W}} D(\boldsymbol{x},\boldsymbol{y}) + \beta\sum_{\boldsymbol{x}\in\mathcal{W}}\sum_{\boldsymbol{y}\in\mathcal{D}} D(\boldsymbol{x},\boldsymbol{y})\\ \text{s.t. } |\mathcal{W}| = k\ .\end{aligned}$$

Since finding the minimum is invariant to scaling, we obtain an equivalent formulation

$$\begin{aligned}\min_{\mathcal{W}\subset\mathcal{D}} \ -\sum_{\boldsymbol{x}\in\mathcal{W}}\sum_{\boldsymbol{y}\in\mathcal{W}} D(\boldsymbol{x},\boldsymbol{y}) + \gamma\sum_{\boldsymbol{x}\in\mathcal{W}}\sum_{\boldsymbol{y}\in\mathcal{D}} D(\boldsymbol{x},\boldsymbol{y})\\ \text{s.t. } |\mathcal{W}| = k\ ,\end{aligned}$$

by setting $\gamma \coloneqq \frac{\beta}{\alpha}$. The parameter γ then weighs the objectives, with a small value preferring the far away objective, while larger values lead to more centralized data points.

Since $\mathcal{W}\subset\mathcal{D}$, we can indicate the membership of the data point $\boldsymbol{x}_i$ in $\mathcal{W}$ by a binary variable z_i. This leads to a QBP formulation by using the distance matrix $\boldsymbol{D} = (D(\boldsymbol{x},\boldsymbol{y}))_{\boldsymbol{x},\boldsymbol{y}\in\mathcal{D}}$

$$\min_{\boldsymbol{z}\in\mathbb{B}^n} \ -\boldsymbol{z}^\top\boldsymbol{D}\boldsymbol{z} + \gamma\left(\boldsymbol{D}\boldsymbol{1}\right)^\top\boldsymbol{z} \tag{4a}$$

$$\text{s.t. } \boldsymbol{1}^\top\boldsymbol{z} = k\ . \tag{4b}$$

A QUBO formulation is obtained [20] by integrating the quadratic penalty function $(\boldsymbol{1}^\top\boldsymbol{z}-k)^2$ into the objective

$$\min_{\boldsymbol{z}\in\mathbb{B}^n} \ -\boldsymbol{z}^\top\boldsymbol{D}\boldsymbol{z} + \gamma\left(\boldsymbol{D}\boldsymbol{1}\right)^\top\boldsymbol{z} + \lambda_{\text{MED}}\left(\boldsymbol{1}^\top\boldsymbol{z}-k\right)^2\ , \tag{5}$$

where equivalence to Eq. (4) holds for large enough $\lambda_{\text{MED}}>0$. Finding a suitable value is far from trivial and is of great interest in current research [30].

Due to $z_iz_i = z_i$ for $z_i\in\mathbb{B}^n$, the linear offset terms in Eq. (5) can be absorbed into a quadratic objective, by adding the corresponding vectors on the diagonal of the quadratic matrix. In other words, the QUBO matrix (see Eq. (1)) of Eq. (5) is given by

$$\boldsymbol{Q}_{\text{MED}} = -\boldsymbol{D} + \lambda_{\text{MED}}\boldsymbol{1}\boldsymbol{1}^\top + \operatorname{diag}\left[\gamma\boldsymbol{D}\boldsymbol{1} - 2\lambda_{\text{MED}}k\boldsymbol{1}\right]\ . \tag{6}$$

B. Kernel Density Estimation

As VQ aims to model the underlying probability density functions of the data distribution with the help of the codebook, we also consider probability density estimates. That is, assuming that $p_{\mathcal{D}}(\cdot)$ is an underlying probability density function of $\mathcal{D}$ and $p_{\mathcal{W}}(\cdot)$ of $\mathcal{W}$, we want $p_{\mathcal{D}}(\cdot)$ and $p_{\mathcal{W}}(\cdot)$ to be as similar as possible. The dissimilarity or *divergence* can be measured in numerous ways, examples include the *Cauchy-Schwartz* divergence [31] or the *Kullback-Leibler* divergence [32].

Since we have no knowledge on the underlying distribution of our data, we use approximations. In particular, we examine *Kernel Density Estimation* (KDE), also called *Parzen windowing*, similar to [21]

$$p_{\mathcal{D}}(\boldsymbol{x}) = \frac{1}{n}\sum_{\boldsymbol{y}\in\mathcal{D}} K(\boldsymbol{x},\boldsymbol{y})\ ,$$

where $K:\mathbb{X}\times\mathbb{X}\to\mathbb{R}$ is a kernel function. In [33, 12], an RBF kernel is used to approximate $p_{\mathcal{D}}(\cdot)$ and $p_{\mathcal{W}}(\cdot)$ and the Cauchy-Schwartz divergence is minimized. However, we are not restricted in the choice of the used kernel. Due to Mercer's theorem [34], there exists a feature map $\phi:\mathbb{X}\to\mathbb{R}^D$, s.t.,

$$p_{\mathcal{D}}(\boldsymbol{x}) = \frac{1}{n}\sum_{\boldsymbol{y}\in\mathcal{D}}\phi(\boldsymbol{x})^\top\phi(\boldsymbol{y}) = \ \phi(\boldsymbol{x})^\top\boldsymbol{\phi}_{\mathcal{X}}, \tag{7a}$$

$$p_{\mathcal{W}}(\boldsymbol{x}) = \frac{1}{k}\sum_{\boldsymbol{y}\in\mathcal{W}}\phi(\boldsymbol{x})^\top\phi(\boldsymbol{y}) = \phi(\boldsymbol{x})^\top\boldsymbol{\phi}_{\mathcal{W}}, \tag{7b}$$

with feature mean vectors

$$\boldsymbol{\phi}_{\mathcal{D}} \coloneqq \frac{1}{n}\sum_{\boldsymbol{x}\in\mathcal{D}}\phi(\boldsymbol{x}),\quad \boldsymbol{\phi}_{\mathcal{W}} \coloneqq \frac{1}{k}\sum_{\boldsymbol{x}\in\mathcal{W}}\phi(\boldsymbol{x})\ .$$

The mean vectors $\boldsymbol{\phi}_{\mathcal{X}}$ and $\boldsymbol{\phi}_{\mathcal{W}}$ fully characterize the KDEs $p_{\mathcal{D}}(\cdot)$ and $p_{\mathcal{W}}(\cdot)$. The difference between these vectors can be measured by the MMD [35], which leads to the following objective

$$\min_{\mathcal{W}} \ \left\|\boldsymbol{\phi}_{\mathcal{D}} - \boldsymbol{\phi}_{\mathcal{W}}\right\|^2 \tag{8a}$$

$$\text{s.t. } |\mathcal{W}| = k\ . \tag{8b}$$

It is worth noting that prototypes coinciding with actual data points are often easier to interpret. This makes KDE-VQ very similar to k-medoids-VQ, and we will investigate this connection later on.

We rewrite the MMD in terms of inner products

$$\begin{aligned}&\min_{\mathcal{W}} \ \left\|\boldsymbol{\phi}_{\mathcal{D}} - \boldsymbol{\phi}_{\mathcal{W}}\right\|^2\\ \Leftrightarrow&\min_{\mathcal{W}} \ \boldsymbol{\phi}_{\mathcal{W}}^\top\boldsymbol{\phi}_{\mathcal{W}} - 2\boldsymbol{\phi}_{\mathcal{D}}^\top\boldsymbol{\phi}_{\mathcal{W}} + \boldsymbol{\phi}_{\mathcal{D}}^\top\boldsymbol{\phi}_{\mathcal{D}}\\ \Leftrightarrow&\min_{\mathcal{W}} \ \frac{1}{k^2}\sum_{\boldsymbol{x}\in\mathcal{W}}\sum_{\boldsymbol{y}\in\mathcal{W}} K(\boldsymbol{x},\boldsymbol{y}) - \frac{2}{kn}\sum_{\boldsymbol{x}\in\mathcal{W}}\sum_{\boldsymbol{y}\in\mathcal{D}} K(\boldsymbol{x},\boldsymbol{y})\ ,\end{aligned}$$

with $K(\boldsymbol{x},\boldsymbol{y}) = \phi(\boldsymbol{x})^\top\phi(\boldsymbol{y})$ since K is a Mercer kernel.

Again, letting the binary variable z_i indicate whether $\boldsymbol{x}_i$ is an element of $\mathcal{W} \subset \mathcal{D}$, we obtain a QBP formulation equivalent to Eq. (8) by multiplying by k^2

$$\min_{\boldsymbol{z}\in\mathbb{B}^n} \ \boldsymbol{z}^\top \boldsymbol{K}\boldsymbol{z} - \frac{2k}{n}(\boldsymbol{K}\mathbf{1})^\top \boldsymbol{z} \tag{9a}$$

$$\text{s.t. } \mathbf{1}^\top \boldsymbol{z} = k \ . \tag{9b}$$

An equivalent QUBO formulation is then given by incorporating a weighted penalty term with large enough $\lambda_{\text{KDE}} > 0$

$$\min_{\boldsymbol{z}\in\mathbb{B}^n} \ \boldsymbol{z}^\top \boldsymbol{K}\boldsymbol{z} - \frac{2k}{n}(\boldsymbol{K}\mathbf{1})^\top \boldsymbol{z} + \lambda_{\text{KDE}} \left(\mathbf{1}^\top \boldsymbol{z} - k\right)^2 \ . \tag{10}$$

Similar to Eq. (6), the QUBO matrix of Eq. (10) is given by

$$\boldsymbol{Q}_{\text{KDE}} = \boldsymbol{K} + \lambda_{\text{KDE}}\mathbf{1}\mathbf{1}^\top - 2\,\text{diag}\left[\frac{k}{n}\boldsymbol{K}\mathbf{1} + \lambda_{\text{MED}} k\mathbf{1}\right] \ . \tag{11}$$

IV. Relationship between MED and KDE

Interestingly, there is a connection between k-medoids-VQ and KDE-VQ. It holds under the mild assumption of using normalized kernel functions.

Proposition 1. *Let $K : \mathbb{X} \times \mathbb{X} \to \mathbb{R}$ be a Mercer kernel and $\phi : \mathbb{X} \to \mathbb{R}^d$ the underlying feature map, that is $\phi(\boldsymbol{x})^\top \phi(\boldsymbol{y}) = K(\boldsymbol{x}, \boldsymbol{y})$. If all feature vectors lie on the unit sphere, i.e., $\|\phi(\boldsymbol{x})\| = 1$, $\forall \boldsymbol{x} \in \mathbb{X}$—also called normalized kernel $K(\boldsymbol{x}, \boldsymbol{x}) = 1$—we can obtain a distance measure $D : \mathbb{X} \times \mathbb{X} \to \mathbb{R}$ by*

$$D(\boldsymbol{x}, \boldsymbol{y}) = 1 - K(\boldsymbol{x}, \boldsymbol{y}) \ . \tag{12}$$

Proof. Consider the squared euclidean distance between feature vectors $\phi(\boldsymbol{x})$ and $\phi(\boldsymbol{y})$

$$\begin{aligned} &\|\phi(\boldsymbol{x}) - \phi(\boldsymbol{y})\|^2 \\ =&\phi(\boldsymbol{x})^\top \phi(\boldsymbol{x}) + \phi(\boldsymbol{y})^\top \phi(\boldsymbol{y}) - 2\phi(\boldsymbol{x})^\top \phi(\boldsymbol{y}) \\ =&2 - 2K(\boldsymbol{x}, \boldsymbol{y}) \ . \end{aligned}$$

Dividing the RHS by 2, we obtain the claim. □

Examples for normalized kernels are RBF kernels, Laplacian kernels or quantum kernels. Using Prop. 1, we first investigate the QUBO formulations.

Theorem 1. *Let $K : \mathbb{X} \times \mathbb{X} \to \mathbb{R}$ be a normalized Mercer kernel, $\mathcal{D} = \{\boldsymbol{x}^1, \ldots, \boldsymbol{x}^n\} \subset \mathbb{X}$, $k < n$ and $\boldsymbol{K}$ be the corresponding kernel matrix. . Defining the matrix $\boldsymbol{D} := \mathbf{1}\mathbf{1}^\top - \boldsymbol{K}$, the QUBO matrices in Eq.* (6) *and Eq.* (11) *are equal if we set $\gamma = \frac{2k}{n}$ and $\lambda_{KDE} = \lambda_{MED} + 1$.*

Proof. Inserting γ and λ_{KDE} into the QUBO matrix definition in Eq. (5) leads to

$$\begin{aligned} &-\boldsymbol{D} + \lambda_{\text{KDE}}\mathbf{1}\mathbf{1}^\top + \text{diag}\left[\gamma \boldsymbol{D}\mathbf{1} - 2\lambda_{\text{KDE}} k\mathbf{1}\right] \\ \Leftrightarrow\ &\boldsymbol{K} + (\lambda_{\text{KDE}} - 1)\mathbf{1}\mathbf{1}^\top + \text{diag}\left[-\gamma \boldsymbol{K}\mathbf{1} - (2\lambda_{\text{KDE}} k - n\gamma)\mathbf{1}\right] \\ \Leftrightarrow\ &\boldsymbol{K} + \lambda_{\text{MED}}\mathbf{1}\mathbf{1}^\top \\ &\quad + \text{diag}\left[-\frac{2k}{n}\boldsymbol{K}\mathbf{1} - \left(2\left(\lambda_{\text{MED}} + 1\right)k - 2\frac{kn}{n}\right)\mathbf{1}\right] \\ \Leftrightarrow\ &\boldsymbol{K} + \lambda_{\text{MED}}\mathbf{1}\mathbf{1}^\top - 2\,\text{diag}\left[\frac{k}{n}\boldsymbol{K}\mathbf{1} + \lambda_{\text{MED}} k\mathbf{1}\right] \ , \end{aligned}$$

which corresponds to the QUBO matrix in Eq. (10). □

The proof of Theorem 1 can be used to show equivalence between the QBP formulations.

Corollary 1. *Let $\boldsymbol{K}$ and $\boldsymbol{D}$ be defined as in Theorem 1. Setting $\gamma = \frac{2k}{n}$, we obtain equivalence between the QBPs given in Eq.* (4) *and Eq.* (9)*.*

Using Theorem 1 and corollary 1, we can see that KDE-VQ is a special case of MED-VQ by choosing parameters in a specific way and using distance measures of the form $D(\boldsymbol{x}, \boldsymbol{y}) = 1 - K(\boldsymbol{x}, \boldsymbol{y})$ for some Mercer kernel. With the commonly used RBF kernel with a bandwidth of 2, one exactly recovers Welsch's M-estimator, which was used as a distance function in [20].

V. Discussion

Our findings highlight a surprising equivalence between the QUBO formulations of MED-VQ and KDE-VQ, revealing a deeper structural relationship that had not been explicitly articulated in prior work. In particular, we showed that under a mild assumption on the kernel's feature map—specifically, that data points are mapped onto the unit sphere—the KDE-VQ QUBO becomes a special case of the MED-VQ QUBO with appropriately chosen parameters.

A key insight arising from this equivalence is the interpretation of the weighting parameter γ in the k-medoids QUBO formulation. Traditionally, γ has been viewed as a heuristic balancing factor, mediating between selecting central and mutually distant data points. Our work refines this perspective by showing that γ can be interpreted as a geometric scaling factor within the kernel-induced feature space. This means that tuning γ effectively scales the geometry of the prototype selection problem: it stretches or compresses the influence of centrality versus diversity in the feature space. In practical terms, small values of γ prioritize selecting prototypes that are maximally distant from each other, while large γ values emphasize proximity to the dataset's density center.

This interpretation not only offers a conceptual understanding of γ but also has implications for the implementation of these QUBO formulations on quantum-inspired or quantum-native hardware. Since QA and Ising machines rely on energy landscapes defined by QUBO matrices, the role of γ translates directly into how the energy landscape is shaped in quantum optimization. That is, it affects the spectral gap size of the problem Hamiltonian, which directly relates to the solvability in terms of needed time in AQC. Thus, carefully choosing γ can be seen as calibrating the kernel-induced geometry to align with the hardware's optimization capabilities, which could be particularly relevant NISQ devices.

In summary, this geometric interpretation of the weighting parameter deepens our understanding of the prototype selection problem in VQ. It also opens avenues for principled parameter tuning and hardware-specific adaptations in future work, bridging the gap between classical, kernel-based clustering and quantum optimization frameworks.

VI. Conclusion

In this work, we demonstrated that the Quadratic Unconstrained Binary Optimization (QUBO) formulation for Kernel Density Estimation-based Vector Quantization (VQ) is a special case of the more general k-medoids-VQ QUBO formulation. This surprising equivalence provides a new lens through which to view prototype selection: as a unified framework that blends distance-based and density-based clustering approaches. By interpreting the weighting parameter as a geometric scaling factor within the kernel-induced feature space, we offer new insights into tuning and interpreting these models—especially in the context of quantum-inspired and quantum-native optimization. These findings pave the way for further exploration of how quantum algorithms and hardware can leverage these insights for efficient and interpretable VQ.

Acknowledgments

This research has been funded by the Federal Ministry of Education and Research of Germany and the state of North-Rhine Westphalia as part of the Lamarr Institute for Machine Learning and Artificial Intelligence.

References

[1] Allen Gersho and Robert M Gray. *Vector quantization and signal compression*. Springer Science & Business Media, 2012.

[2] Teuvo Kohonen and Teuvo Kohonen. "Learning vector quantization". In: *Self-Organizing Maps* (2001), pp. 245–261.

[3] Paulo JG Lisboa et al. "The coming of age of interpretable and explainable machine learning models". In: *Neurocomputing* 535 (2023), pp. 25–39.

[4] Alfredo Vellido. "The importance of interpretability and visualization in machine learning for applications in medicine and health care". In: *Neural Computing and Applications* 32.24 (2020), pp. 18069–18083.

[5] Jinkyu Kim and John Canny. "Interpretable learning for self-driving cars by visualizing causal attention". In: *Proceedings of the 2017 International Conference on Computer Vision (ICCV)*. IEEE, 2017, pp. 2942–2950.

[6] Vahe Tshitoyan et al. "Unsupervised word embeddings capture latent knowledge from materials science literature". In: *Nature* 571.7763 (2019), pp. 95–98.

[7] Yizong Cheng. "Mean shift, mode seeking, and clustering". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 17.8 (1995), pp. 790–799.

[8] James MacQueen. "Some methods for classification and analysis of multivariate observations". In: *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*. Vol. 5. University of California Press. 1967, pp. 281–298.

[9] Stuart P Lloyd. "Least squares quantization in PCM". In: *IEEE Transactions on Information Theory* 28.2 (1982), pp. 129–137.

[10] Leonard Kaufman and Peter J Rousseeuw. "Clustering by means of medoids". In: *Proceedings of the First International Conference on Statistical Data Analysis Based on the L1-Norm* (1987), pp. 405–416.

[11] Sudhir Rao, Seungju Han, and Jose Principe. "Information theoretic vector quantization with fixed point updates". In: *2007 International Joint Conference on Neural Networks (IJCNN)*. IEEE. 2007, pp. 1020–1024.

[12] Jian-Wu Xu et al. "A reproducing kernel Hilbert space framework for information-theoretic learning". In: *IEEE Transactions on Signal Processing* 56.12 (2008), pp. 5891–5902.

[13] Daniel Aloise et al. "NP-hardness of Euclidean sum-of-squares clustering". In: *Machine learning* 75 (2009), pp. 245–248.

[14] Abraham P Punnen. *The quadratic unconstrained binary optimization problem*. Springer, 2022.

[15] Naeimeh Mohseni, Peter L McMahon, and Tim Byrnes. "Ising machines as hardware solvers of combinatorial optimization problems". In: *Nature Reviews Physics* 4.6 (2022), p. 363.

[16] Michael A Nielsen and Isaac L Chuang. *Quantum computation and quantum information*. Cambridge University Press, 2010.

[17] Tadashi Kadowaki and Hidetoshi Nishimori. "Quantum annealing in the transverse Ising model". In: *Physical Review E* 58.5 (1998), p. 5355.

[18] Tameem Albash and Daniel A Lidar. "Adiabatic quantum computation". In: *Reviews of Modern Physics* 90.1 (2018), p. 015002.

[19] John Preskill. "Quantum computing in the NISQ era and beyond". In: *Quantum* 2 (2018), p. 79.

[20] Christian Bauckhage et al. "A QUBO formulation of the k-medoids problem". In: *Proceedings of the Conference on "Lernen, Wissen, Daten, Analysen" (LWDA)*. CEUR-WS.org, 2019, p. 54.

[21] C. Bauckhage, R. Ramamurthy, and R. Sifa. "Hopfield networks for vector quantization". In: *Proceedings of the 29th International Conference on Artificial Neural Networks (ICANN)*. Springer, 2020, p. 192.

[22] DJ Laughhunn. "Quadratic binary programming with application to capital-budgeting problems". In: *Operations research* 18.3 (1970), p. 454.

[23] Vojtěch Havlíček et al. "Supervised learning with quantum-enhanced feature spaces". In: *Nature* 567.7747 (2019), pp. 209–212.

[24] Thomas Villmann and Alexander Engelsberger. "Quantum-hybrid neural vector quantization–a mathematical approach". In: *Proceedings of the 21st International Conference on Artificial Intelligence and Soft Computing (ICAISC)*. Springer. 2021, pp. 246–257.

[25] Alexander Engelsberger, Ronny Schubert, and Thomas Villmann. "Steps forward to quantum learning vector quantization for classification learning on a theoretical quantum computer". In: *Proceedings of the 14th Inter-*

national Workshop on Self-Organizing Maps (WSOM). Springer. 2022, pp. 63–73.

[26] Thomas Villmann et al. "Quantum-Inspired Learning Vector Quantization for Classification Learning." In: *Proceedings of the 28th European Symposium on Artificial Neural Networks (ESANN)*. 2020, pp. 279–284.

[27] Thom Sijpesteijn and Frank Phillipson. "Quantum approaches for medoid clustering". In: *Proceedings of the 23rd International Conference on Innovations for Community Services (I4CS)*. Springer. 2023, pp. 222–235.

[28] Alexander Engelsberger and Thomas Villmann. "Quantum-ready vector quantization: Prototype learning as a binary optimization problem". In: *Proceedings of the 31st European Symposium on Artificial Neural Networks (ESANN)*. 2023.

[29] Ronald A Fisher. "On the" probable error" of a coefficient of correlation deduced from a small sample". In: *Metron* 1 (1921), pp. 3–32.

[30] Edoardo Alessandroni et al. "Alleviating the quantum Big-M problem". In: *arXiv preprint arXiv:2307.10379* (2023).

[31] Robert Jenssen et al. "The Cauchy–Schwarz divergence and Parzen windowing: Connections to graph theory and Mercer kernels". In: *Journal of the Franklin Institute* 343.6 (2006), p. 614.

[32] Solomon Kullback and Richard A Leibler. "On information and sufficiency". In: *The Annals of Mathematical Statistics* 22.1 (1951), p. 79.

[33] Tue Lehn-Schiøler et al. "Vector quantization using information theoretic concepts". In: *Natural Computing* 4.1 (2005), p. 39.

[34] James Mercer. "Functions of positive and negative type, and their connection the theory of integral equations". In: *Philosophical Transactions of the Royal Society of London* 209 (1909), p. 415.

[35] Arthur Gretton et al. "A kernel two-sample test". In: *The Journal of Machine Learning Research* 13.1 (2012), p. 723.