METRIC-ADAPTIVE ONE-CLASS CLASSIFICATION WITH GENERALIZED LEARNING VECTOR QUANTIZATION

Nana Abeka Otoo
Authentic-Network
Chemnitz, Germany
nana.abekaotoo@authentic.network

Asirifi Boa
Independent Researcher
Colorado, USA
boaasirifi@gmail.com

ABSTRACT

Prototype based one-class learners such as Supervised Vector Quantization for One-Class Classification (SVQ-OCC) utilise squared Euclidean distance without metric adaptation and cannot adapt to the spatial structure of the data. Moreover, these learners are unable to capture relevant variation when the target distribution presents anisotropic structure in the feature space. This paper proposes One-Class Generalized Learning Vector Quantization (OC-GLVQ), whose discriminant substitutes the negative-class distance in GLVQ with a per-prototype visibility threshold, yielding a cost function that is jointly differentiable in prototypes, thresholds and metric parameters. A hierarchy of distance formulations from diagonal relevance weighting through global and local matrix projections to tangent subspace metrics is developed within the OC-GLVQ framework. The resulting classifiers adapt their geometry to the target distribution while retaining prototype-level interpretability. This paper identifies a gradient-magnitude mismatch between the prototypes and threshold parameters in high-dimensional settings and resolves it through a square-root reparameterisation. Experimental assessment on four practical datasets indicates the metric-adaptive variants consistently surpass both SVQ-OCC and Support Vector Data Description (SVDD), with local matrix adaptation yielding the strongest overall performance.

Keywords: one-class classification, learning vector quantization, metric adaptation, prototype-based learning, generalized matrix LVQ

1 Introduction

One-Class Classification (OCC) addresses the learning task of identifying whether a test instance belongs to a target class, given that the available training data predominantly represents that class [1, 2]. Unlike anomaly detection, which assumes contaminated training data, OCC assumes that the target class is well-characterised and aims to reject all variations from its learned description [4]. Applications arise wherever anomalous observations are rare, expensive to collect and inherently unbounded in variety: industrial fault detection, medical screening and network intrusion analysis all present the OCC structure [3]. The classical approach to OCC via Support Vector Data Description (SVDD) [1] encloses target data in a minimum-volume hypersphere in a kernel-induced feature space. The related One-Class Support Vector Machine (OC-SVM) [12] separates target data from the origin by a maximum-margin hyperplane. Deep SVDD [13] enhances the formulation by jointly learning a neural network mapping and the enclosing hypersphere. These models are effective but provide no prototype-level interpretability of the learned decision boundary.

Prototype based classifiers offer an interpretable formulation for OCC. A set of prototypes represents the target distribution and the distance between a test sample and its nearest prototype determines class membership. SVQ-OCC [5] demonstrated solid performance against SVDD on several benchmarks, but relies on squared Euclidean distances throughout (see Section 2 for details). Angiulli [16] proposed a prototype based domain description that characterises the target boundary through distance profiles, likewise without metric adaptation. Fischer, Hammer and Wersing [15] developed rejection strategies for prototype classifiers that extend to one-class settings, although their focus lies on classification confidence rather than metric learning. None of these methods adapts the distance function to the structure of the target data. Adapting the distance function to the data has a long history in supervised Learning Vector

Quantization (LVQ) [17, 18, 19]. A hierarchy of learnable metrics has been developed, from diagonal feature weighting (Generalized Relevance LVQ [8]) through full and per-prototype linear projections (Generalized Matrix LVQ and its localized variant LGMLVQ [9]) to tangent subspace distances (Generalized Tangent LVQ [10]), substantially boosting classification on data with considerable correlation structure [20, 14]. Analogous goals are pursued in the more extensive metric learning literature [21]. In the one-class setting, all prototypes belong to the target class and no wrong-class prototype exists. The Generalized LVQ (GLVQ) discriminant [7] is adapted by replacing the negative-class distance with a learned threshold per prototype (Section 3). However, the interaction between learned metrics and the threshold parameter has not been investigated.

This paper formulates metric-adaptive variants of OC-GLVQ, namely OC-GRLVQ, OC-GMLVQ, OC-LGMLVQ and OC-GTLVQ, presenting a systematic hierarchy from diagonal to subspace metrics for one-class prototype learning. Each model's cost function, distance gradients and parameter adaptation rules are derived in full in a unified gradient framework. In the development of the GLVQ-based OC-variants, a gradient-magnitude mismatch between the prototypes and threshold parameters is identified and resolved through a square-root reparameterisation $\theta_k = \alpha_k^2$. Experimental evaluation through 10-fold cross-validation (CV) on four datasets indicates that the metric-adaptive OC-GLVQ variants outperform both SVQ-OCC and SVDD by wide margins in F_1 score. Section 2 reviews GLVQ with adaptive metrics and OCC with prototypes. Section 3 presents the OC-GLVQ discriminant with its gradient mismatch analysis and the metric extensions with their adaptation rules. Section 4 describes the experimental protocol and Section 5 reports on results. Section 6 interprets the findings and Section 7 concludes.

2 Learning Vector Quantization and One-Class Classification

Let $\mathbf{S} = \{\mathbf{s}_1, \ldots, \mathbf{s}_N\} \subset \mathbb{R}^m$ be a training set with class labels $c_\mathbf{s} \in \{\oplus, \ominus\}$ distinguishing target from non-target observations. The target class is represented by a codebook $\mathbf{W} = \{\mathbf{w}_1, \ldots, \mathbf{w}_K\} \subset \mathbb{R}^m$ of prototypes, each carrying a visibility threshold $\theta_k > 0$. A test instance $\mathbf{s}$ is classified as target whenever its distance to the closest prototype remains below that prototype's threshold; otherwise it is flagged as an outlier. The threshold based decision rule builds on the GLVQ framework [7], whose standard multi-class cost aggregates a normalised margin over the training set:

$$E_{\mathrm{GLVQ}} = \sum_{\mathbf{s}\in\mathbf{S}} f_\beta(\mu(\mathbf{s})), \qquad \mu(\mathbf{s}) = \frac{d(\mathbf{s}, \mathbf{w}^+) - d(\mathbf{s}, \mathbf{w}^-)}{d(\mathbf{s}, \mathbf{w}^+) + d(\mathbf{s}, \mathbf{w}^-)}, \tag{1}$$

with

$$\mathbf{w}^+ = \underset{\mathbf{w}_r:\, c_r = c_\mathbf{s}}{\arg\min}\, d(\mathbf{s}, \mathbf{w}_r), \qquad \mathbf{w}^- = \underset{\mathbf{w}_r:\, c_r \neq c_\mathbf{s}}{\arg\min}\, d(\mathbf{s}, \mathbf{w}_r),$$

where $\mathbf{w}_r \in \mathbf{W}$ denotes a candidate prototype with label c_r, $\mathbf{w}^+$ is the nearest prototype whose label matches that of $\mathbf{s}$ and $\mathbf{w}^-$ is the nearest prototype with a different label. The margin $\mu(\mathbf{s}) \in [-1, +1]$ is negative when the sample falls closer to the correct-class prototype than to any competitor and positive otherwise. Passing it through a monotonically increasing transfer function f_β produces a differentiable cost; the logistic sigmoid $f_\beta(x) = (1 + \exp(-\beta x))^{-1}$ with steepness $\beta > 0$ is the standard choice [7]. The differentiable distance d in (1) is not restricted to the Euclidean norm. When discriminative information is concentrated on a subset of features and in inter-feature correlations, a fixed isotropic metric is insufficient. Several learnable replacements have been proposed, forming a hierarchy of increasing geometric expressiveness [20]. The simplest variant with relevances is GRLVQ [8], where the feature dimensions are scaled by a learned non-negative weight:

$$d_\lambda(\mathbf{s}, \mathbf{w}) = \sum_{j=1}^{m} \lambda_j \, (s_j - w_j)^2, \qquad \lambda_j \geq 0, \quad \sum_j \lambda_j = 1. \tag{2}$$

The normalisation $\sum_j \lambda_j = 1$ prevents trivial growth of the weights. After training, λ_j directly indicates how much feature j contributes to classification; dimensions assigned $\lambda_j \approx 0$ are effectively discarded. GMLVQ [9] generalises the diagonal relevance matrix to a full linear projection $\Omega \in \mathbb{R}^{l\times m}$, measuring distances in the resulting transformed space:

$$d_\Omega(\mathbf{s}, \mathbf{w}) = \|\Omega(\mathbf{s} - \mathbf{w})\|^2 = (\mathbf{s} - \mathbf{w})^T \Lambda (\mathbf{s} - \mathbf{w}), \tag{3}$$

where the positive semi-definite tensor $\Lambda = \Omega^T\Omega$ is kept normalised at $\mathrm{tr}(\Lambda) = m$. All prototypes share a single Ω in GMLVQ; the local variant LGMLVQ removes the shared Ω constraint by giving every prototype its own projection Ω_k, so that different regions of the feature space are characterised according to distinct geometries. An altogether different strategy is taken by GTLVQ [10], which instead of amplifying discriminative directions seeks to identify and remove invariant ones. Each prototype carries an orthonormal basis $\Psi_k \in \mathbb{R}^{m\times p}$ spanning its local invariance subspace and the distance is defined as

$$d_T(\mathbf{s}, \mathbf{w}_k) = \|\boldsymbol{\delta}_k\|^2 - \|\Psi_k^T \boldsymbol{\delta}_k\|^2, \qquad \boldsymbol{\delta}_k = \mathbf{s} - \mathbf{w}_k. \tag{4}$$

The subtraction removes the squared projection onto the invariance subspace, so that only the displacement component orthogonal to learned invariances contributes to the distance. OCC employs all prototypes to represent the target class and no negative-class prototype exists. Two established approaches for prototype-based OCC are SVQ-OCC [5] and the threshold-based discriminant formulation. SVQ-OCC combines a Neural Gas [6] representation cost with a classification cost based on probabilistic response functions. Each prototype $\mathbf{w}_k$ has a visibility range θ_k and a response function $f(d(\mathbf{s}, \mathbf{w}_k), \theta_k)$ that decays monotonically with distance. The unit step function $H(\theta_k - d(\mathbf{s}, \mathbf{w}_k))$ restricts the response beyond θ_k and a contrastive score drives learning of prototypes and visibility ranges simultaneously. The threshold based formulation modifies the GLVQ discriminant directly by replacing $d(\mathbf{s}, \mathbf{w}^-)$ in (1) with a learned parameter θ_k associated with the nearest prototype. The replacement of $d(\mathbf{s}, \mathbf{w}^-)$ with θ_k produces a discriminant that stays bounded in $[-1, +1]$ and admits gradient-based optimisation of prototypes and thresholds jointly. SVDD [1] seeks the smallest hypersphere in a kernel-induced feature space that encloses the target data. A parameter $\nu \in (0, 1]$ controls the fraction of permitted outliers among target samples and the decision boundary is the sphere surface.

3 Metric-Adaptive One-Class GLVQ

The OC-GLVQ discriminant is now extended to metric-adaptive variants, namely OC-GRLVQ, OC-GMLVQ, OC-LGMLVQ and OC-GTLVQ. In every variant, the cost function preserves the structure of (7) where only the distance function d and the associated learnable metric parameters differ. The displacement vector $\boldsymbol{\delta}_k = \mathbf{s} - \mathbf{w}_k$ from prototype $\mathbf{w}_k$ to sample $\mathbf{s}$ is used throughout.

3.1 OC-GLVQ Discriminant and Cost Function

Given an input sample $\mathbf{s}$, the nearest prototype under distance d is

$$k^*(\mathbf{s}) = \underset{k \in \{1,\ldots,K\}}{\arg\min}\ d(\mathbf{s}, \mathbf{w}_k). \tag{5}$$

Each prototype $\mathbf{w}_k$ has a visibility threshold $\theta_k > 0$ that defines the boundary of its acceptance region. The signed discriminant for sample $\mathbf{s}$ is

$$\mu(\mathbf{s}) = \xi(\mathbf{s}) \cdot \frac{d(\mathbf{s}, \mathbf{w}_{k^*}) - \theta_{k^*}}{d(\mathbf{s}, \mathbf{w}_{k^*}) + \theta_{k^*} + \varepsilon}, \tag{6}$$

where $\xi(\mathbf{s}) = +1$ if $c_\mathbf{s} = \oplus$ and $\xi(\mathbf{s}) = -1$ if $c_\mathbf{s} = \ominus$. The constant $\varepsilon > 0$ ensures numerical stability. The discriminant takes values in $(-1, +1)$ with $\mu < 0$ indicating correct classification. Hence for target samples, $\mu < 0$ when $d < \theta$ (sample lies within the visibility range) and for non-target samples, $\mu < 0$ when $d > \theta$ (sample lies outside). The cost function to be optimised is

$$E = \frac{1}{N} \sum_{i=1}^{N} f_\beta(\mu(\mathbf{s}_i) + \gamma), \tag{7}$$

where $f_\beta(z) = (1 + \exp(-\beta z))^{-1}$ is the logistic sigmoid with steepness $\beta > 0$ and $\gamma \geq 0$ is a margin parameter. The margin γ shifts the decision boundary, requiring that correctly classified samples satisfy $\mu < -\gamma$ rather than merely $\mu < 0$. The shift penalises samples near the boundary and encourages well-separated prototypes. The gradient of the cost with respect to any parameter ϕ in the model factorises as

$$\frac{\partial E}{\partial \phi} = \frac{1}{N} \sum_{i=1}^{N} \underbrace{\beta\, f_\beta(\mu_i + \gamma)(1 - f_\beta(\mu_i + \gamma))}_{= f'_\beta(\mu_i + \gamma)} \cdot \frac{\partial \mu_i}{\partial \phi}, \tag{8}$$

where $\mu_i = \mu(\mathbf{s}_i)$. The classifier function derivatives with respect to distance and threshold are

$$\xi_k^+ = \frac{\partial \mu}{\partial d} = \xi \cdot \frac{2\theta}{(d + \theta)^2}, \qquad \xi_k^- = \frac{\partial \mu}{\partial \theta} = \xi \cdot \frac{-2d}{(d + \theta)^2}, \tag{9}$$

where $d = d(\mathbf{s}, \mathbf{w}_{k^*})$ and $\theta = \theta_{k^*}$ are used. These factors mediate the distance and threshold contributions to the cost gradient respectively. For the base OC-GLVQ model with squared Euclidean distance $d = \|\boldsymbol{\delta}_k\|^2$, the resulting prototype and threshold gradients are

$$\frac{\partial E}{\partial \mathbf{w}_k} = f'_\beta \cdot \xi_k^+ \cdot (-2\boldsymbol{\delta}_k), \tag{10}$$

$$\frac{\partial E}{\partial \theta_k} = f'_\beta \cdot \xi_k^-. \tag{11}$$

The resulting update rules with learning rate $\eta > 0$ are

$$\mathbf{w}_k \leftarrow \mathbf{w}_k - \eta \cdot f'_\beta \cdot \xi_k^+ \cdot (-2\boldsymbol{\delta}_k), \tag{12}$$

$$\theta_k \leftarrow \theta_k - \eta \cdot f'_\beta \cdot \xi_k^-. \tag{13}$$

The prototype update in (12) displaces $\mathbf{w}_k$ along the direction determined by the sign of ξ_k^+, which encodes both the class label and the relative magnitudes of distance and threshold. The update of the threshold in (13) adjusts the acceptance region according to the class label. For target samples ($\xi = +1$), ξ_k^- is negative, so gradient descent increases θ_k, expanding the visibility range to encompass target data. For non-target samples ($\xi = -1$), ξ_k^- is positive and hence gradient descent decreases θ_k, contracting the visibility range to exclude outliers. The balance between these two forces determines the learned threshold. The factors ξ_k^+ and ξ_k^- diminish with increasing $d + \theta$ due to the quadratic denominator, so samples far from the decision boundary receive attenuated gradient signal.

3.2 Gradient Mismatch in Threshold Learning

We consider a comparison of the gradient magnitudes of different parameter groups as the input dimensionality increases. Considering squared Euclidean distance in $\mathbb{R}^m$, the distance $d(\mathbf{s}, \mathbf{w}_k) = \|\boldsymbol{\delta}_k\|^2$ scales as $\mathcal{O}(m)$ when feature values have unit variance. The threshold θ_k must balance the scale $\mathcal{O}(m)$ to induce relevant discriminant values. When $d, \theta \sim \mathcal{O}(m)$, the denominator $(d + \theta)^2 \sim \mathcal{O}(m^2)$ and both partial derivatives in (9) scale as $\mathcal{O}(1/m)$. Regarding the prototype gradient, the chain rule gives

$$\frac{\partial E}{\partial \mathbf{w}_k} = f'_\beta \cdot \frac{\partial \mu}{\partial d} \cdot \frac{\partial d}{\partial \mathbf{w}_k}, \tag{14}$$

where $\partial d / \partial \mathbf{w}_k = -2\boldsymbol{\delta}_k$ has magnitude $\mathcal{O}(\sqrt{m})$. The gradient magnitude of the resulting prototype is $\mathcal{O}(1/m) \cdot \mathcal{O}(\sqrt{m}) = \mathcal{O}(1/\sqrt{m})$. The threshold gradient lacks the amplifier effect:

$$\left| \frac{\partial E}{\partial \theta_k} \right| \sim \mathcal{O}(1/m). \tag{15}$$

Proposition 1 (Gradient Mismatch)**.** *Let the features be standardised to zero mean and unit variance so that $\mathbb{E}[\delta_{k,j}^2] = \mathcal{O}(1)$ for each component and let the prototypes lie within $\mathcal{O}(\sqrt{m})$ Euclidean distance of the data. Under squared Euclidean distance in $\mathbb{R}^m$, the prototype gradient magnitude exceeds the threshold gradient magnitude by a factor of $\sqrt{m}$:*

$$\frac{\|\partial E / \partial \mathbf{w}_k\|}{|\partial E / \partial \theta_k|} \sim \sqrt{m}. \tag{16}$$

The slow adaptation of the threshold with respect to the prototypes is a result of the observed imbalances, specifically in high-dimensional input spaces. An adaptive optimiser such as Adam [11] partially compensates through per-parameter learning rate scaling, but the fundamental mismatch remains in the signal-to-noise ratio of the threshold gradient.

3.3 Square-Root Reparameterisation

The mismatch identified in Section 3.2 is fixed by restoring balanced gradient magnitudes. To this end, the threshold is reparameterised through its square root. Let $\alpha_k \in \mathbb{R}$ with $\theta_k = \alpha_k^2$. The gradient with respect to α_k is

$$\frac{\partial E}{\partial \alpha_k} = \frac{\partial E}{\partial \theta_k} \cdot 2\alpha_k. \tag{17}$$

Since $\alpha_k = \sqrt{\theta_k} \sim \mathcal{O}(\sqrt{m})$, the magnitude becomes $\mathcal{O}(1/m) \cdot \mathcal{O}(\sqrt{m}) = \mathcal{O}(1/\sqrt{m})$, equalising the prototype gradient magnitude exactly.

Theorem 1 (Gradient Balance)**.** *Under the reparameterisation $\theta_k = \alpha_k^2$, both prototype and threshold gradients scale as $\mathcal{O}(1/\sqrt{m})$, eliminating the $\sqrt{m}$ imbalance.*

Proof. The prototype gradient magnitude is $|f'_\beta| \cdot |\xi_k^+| \cdot \|\partial d / \partial \mathbf{w}_k\| \sim \mathcal{O}(1/m) \cdot \mathcal{O}(\sqrt{m}) = \mathcal{O}(1/\sqrt{m})$. The reparameterised threshold gradient magnitude is $|f'_\beta| \cdot |\xi_k^-| \cdot 2|\alpha_k| \sim \mathcal{O}(1/m) \cdot \mathcal{O}(\sqrt{m}) = \mathcal{O}(1/\sqrt{m})$. The ratio is $\mathcal{O}(1)$, independent of m. □

Additionally, the squaring ensures $\theta_k \geq 0$ without explicit clamping, providing an unconstrained parameterisation of the non-negative threshold. The parameter α_k is initialised from the mean Voronoi distance of target samples. Let $\mathbf{S}_k = \{\mathbf{s} \in \mathbf{S}_\oplus : k^*(\mathbf{s}) = k\}$ be the Voronoi cell of prototype $\mathbf{w}_k$ restricted to target data. Then

$$\alpha_k = \sqrt{\max\left(\frac{1}{|\mathbf{S}_k|}\sum_{\mathbf{s}\in\mathbf{S}_k} d(\mathbf{s}, \mathbf{w}_k),\ \varepsilon\right)}, \tag{18}$$

where $\varepsilon > 0$ prevents degenerate initialisation when a Voronoi cell is empty and d denotes the squared Euclidean distance regardless of the metric variant used during training. The initialisation (18) sets the visibility range to the root mean square distance of assigned target samples, providing a data-driven starting point for threshold learning. The metric-specific distance is not used for initialisation because the metric parameters ($\boldsymbol{\lambda}$, Ω, Ψ_k) are themselves uninformed at initialisation.

3.4 OC-GRLVQ: Diagonal Relevance Adaptation

The cost function (7) and reparameterisation (17) are independent of the choice of distance d, which is now extended to learnable metrics. The simplest step beyond Euclidean distance is a diagonal weighting of features. OC-GRLVQ extends the base OC-GLVQ discriminant by replacing the Euclidean distance with the relevance-weighted distance (2), following the relevance learning paradigm introduced by Hammer and Villmann [8]. The relevance vector is parameterised through a softmax to ensure non-negativity and normalisation:

$$\lambda_j = \frac{\exp(\rho_j)}{\sum_{l=1}^{m}\exp(\rho_l)}, \quad j = 1, \ldots, m, \tag{19}$$

where $\boldsymbol{\rho} \in \mathbb{R}^m$ are unconstrained parameters. The resulting cost function is

$$E_{\text{OC-GRLVQ}}(\mathbf{W}, \boldsymbol{\alpha}, \boldsymbol{\rho}) = \frac{1}{N}\sum_{\mathbf{s}\in\mathbf{S}} f_\beta(\mu^\lambda(\mathbf{s}) + \gamma), \tag{20}$$

with μ^λ evaluated under the diagonal relevance distance (2). The learnable parameters are the prototypes $\mathbf{W}$, the reparameterised thresholds $\boldsymbol{\alpha}$ and the unconstrained relevance vector $\boldsymbol{\rho}$. All other components (the signed label ξ, the threshold $\theta_{k^*} = \alpha_{k^*}^2$ and the cost structure (7)) remain unchanged. Writing $\boldsymbol{\delta}_k = \mathbf{s} - \mathbf{w}_k$, the distance gradients are

$$\frac{\partial d_\lambda}{\partial w_{k,j}} = -2\lambda_j\,\delta_{k,j}, \qquad \frac{\partial d_\lambda}{\partial \lambda_j} = \delta_{k,j}^2. \tag{21}$$

Substituting into the general gradient framework (8) yields the prototype gradient:

$$\frac{\partial E}{\partial w_{k,j}} = f'_\beta \cdot \xi_k^+ \cdot (-2\lambda_j)\,\delta_{k,j}. \tag{22}$$

The resulting update rule is

$$w_{k,j} \leftarrow w_{k,j} - \eta \cdot f'_\beta \cdot \xi_k^+ \cdot (-2\lambda_j)\,\delta_{k,j}. \tag{23}$$

The relevance weighting λ_j rescales the update along each coordinate, so that features deemed irrelevant by the model receive attenuated prototype corrections. The gradient with respect to the unconstrained relevance parameters requires the full softmax Jacobian:

$$\frac{\partial E}{\partial \rho_j} = f'_\beta \cdot \xi_k^+ \cdot \sum_{l=1}^{m} \delta_{k,l}^2 \cdot \frac{\partial \lambda_l}{\partial \rho_j}, \tag{24}$$

where $\partial\lambda_j/\partial\rho_j = \lambda_j(1-\lambda_j)$ and $\partial\lambda_l/\partial\rho_j = -\lambda_l\lambda_j$ for $l \neq j$. The resulting relevance update rule is

$$\rho_j \leftarrow \rho_j - \eta \cdot f'_\beta \cdot \xi_k^+ \cdot \lambda_j\big(\delta_{k,j}^2 - d_\lambda\big)\,. \tag{25}$$

The softmax Jacobian links all relevance parameters, so increasing the weight of one feature will decrease the weights of the others. The softmax parameterisation keeps the normalisation constraint $\sum_j \lambda_j = 1$ in place without needing projection. After training, the learned $\boldsymbol{\lambda}$ exhibits the features that are most relevant regarding the description of the target class. If $\lambda_j \approx 0$, feature j does not provide useful information for OCC. One practical issue with the softmax parameterisation is gradient saturation. When a weight λ_j gets close to one, the Jacobian entries $\lambda_j(1-\lambda_j)$ get close to zero and the gradient signal disappears for all relevance parameters. In practice, such saturation does not happen often. The Adam optimiser's moment normalisation helps offset the effect of shrinking gradients. However, if the relevance profile is very unbalanced in low-dimensional problems, it may need to be monitored.

3.5 OC-GMLVQ: Global Matrix Adaptation

Diagonal relevance cannot capture correlations between features. The next step in the metric hierarchy is a full linear transformation. OC-GMLVQ replaces the distance with the matrix-projected distance (3), using a global transformation $\Omega \in \mathbb{R}^{l \times m}$ shared across all prototypes, as proposed for the supervised setting by Schneider et al. [9]. In the full-rank case, $l = m$ and $\Lambda = \Omega^T \Omega$ is a full $m \times m$ positive semi-definite matrix. The resulting cost function is

$$E_{\text{OC-GMLVQ}}(\mathbf{W}, \boldsymbol{\alpha}, \Omega) = \frac{1}{N} \sum_{\mathbf{s} \in \mathbf{S}} f_\beta(\mu^\Omega(\mathbf{s}) + \gamma), \tag{26}$$

with μ^Ω evaluated under the matrix distance (3). The learnable parameters are the prototypes, the reparameterised thresholds and a single global transformation $\Omega \in \mathbb{R}^{l \times m}$. The distance gradients are

$$\frac{\partial d_\Omega}{\partial \mathbf{w}_k} = -2\,\Lambda\,\boldsymbol{\delta}_k, \qquad \frac{\partial d_\Omega}{\partial \Omega} = 2\,\Omega\,\boldsymbol{\delta}_k \boldsymbol{\delta}_k^T, \tag{27}$$

with $\Lambda = \Omega^T \Omega$ and $\boldsymbol{\delta}_k = \mathbf{s} - \mathbf{w}_k$. Substituting into the general gradient framework (8) gives the prototype and matrix gradients:

$$\frac{\partial E}{\partial \mathbf{w}_k} = f'_\beta \cdot \xi_k^+ \cdot (-2\,\Lambda\,\boldsymbol{\delta}_k), \tag{28}$$

$$\frac{\partial E}{\partial \Omega} = f'_\beta \cdot \xi_k^+ \cdot 2\,\Omega\,\boldsymbol{\delta}_k \boldsymbol{\delta}_k^T. \tag{29}$$

The resulting update rules are

$$\mathbf{w}_k \leftarrow \mathbf{w}_k - \eta \cdot f'_\beta \cdot \xi_k^+ \cdot (-2\,\Lambda\,\boldsymbol{\delta}_k), \tag{30}$$

$$\Omega \leftarrow \Omega - \eta \cdot f'_\beta \cdot \xi_k^+ \cdot 2\,\Omega\,\boldsymbol{\delta}_k \boldsymbol{\delta}_k^T. \tag{31}$$

The prototype update in (30) adjusts prototypes in directions weighted by the learned metric tensor $\Lambda = \Omega^T \Omega$, directing corrections toward features identified as relevant for target characterisation. The matrix update in (31) constitutes a rank-one perturbation of Ω, scaled by the outer product $\boldsymbol{\delta}_k \boldsymbol{\delta}_k^T$. The effect is to rotate the projection and increase the distance along the current displacement direction. Following each gradient step, Ω is rescaled to ensure $\operatorname{tr}(\Omega^T \Omega) = m$:

$$\Omega \leftarrow \Omega \cdot \sqrt{\frac{m}{\operatorname{tr}(\Omega^T \Omega)}}. \tag{32}$$

The rescaling keeps the matrix well-conditioned, avoiding both degeneracy and divergence while preserving the orientation established by the learned metric.

3.6 OC-LGMLVQ: Per-Prototype Local Matrix Adaptation

A shared global metric imposes a single geometry on the entire target distribution, which is overly restrictive when the target class is multimodal with heterogeneous local structure. OC-LGMLVQ addresses the restriction by assigning a private projection $\Omega_k \in \mathbb{R}^{l \times m}$ to each prototype $\mathbf{w}_k$ [9], so that different regions of the target distribution are characterised under distinct metric geometries. The resulting cost function is

$$E_{\text{OC-LGMLVQ}}(\mathbf{W}, \boldsymbol{\alpha}, \{\Omega_k\}) = \frac{1}{N} \sum_{\mathbf{s} \in \mathbf{S}} f_\beta(\mu^{\Omega_k}(\mathbf{s}) + \gamma), \tag{33}$$

with μ^{Ω_k} evaluated under the local metric of the nearest prototype. The learnable parameters are the prototypes, the reparameterised thresholds and K transformation matrices $\{\Omega_k\}_{k=1}^K$. Differentiating the local distance with respect to each parameter gives

$$\frac{\partial d_k}{\partial \mathbf{w}_k} = -2\,\Lambda_k\,\boldsymbol{\delta}_k, \qquad \frac{\partial d_k}{\partial \Omega_k} = 2\,\Omega_k\,\boldsymbol{\delta}_k \boldsymbol{\delta}_k^T, \tag{34}$$

with $\Lambda_k = \Omega_k^T \Omega_k$. The resulting prototype gradient is

$$\frac{\partial E}{\partial \mathbf{w}_k} = f'_\beta \cdot \xi_k^+ \cdot (-2\,\Lambda_k\,\boldsymbol{\delta}_k). \tag{35}$$

Only the nearest prototype's matrix receives gradient signal for a given sample:

$$\frac{\partial E}{\partial \Omega_k} = \begin{cases} f'_\beta \cdot \xi_k^+ \cdot 2\,\Omega_k\,\boldsymbol{\delta}_k \boldsymbol{\delta}_k^T, & \text{if } k = k^*, \\ 0, & \text{otherwise.} \end{cases} \tag{36}$$

The update rules are

$$\mathbf{w}_k \leftarrow \mathbf{w}_k - \eta \cdot f'_\beta \cdot \xi_k^+ \cdot (-2\,\Lambda_k\,\boldsymbol{\delta}_k), \tag{37}$$

$$\Omega_k \leftarrow \Omega_k - \eta \cdot f'_\beta \cdot \xi_k^+ \cdot 2\,\Omega_k\,\boldsymbol{\delta}_k \boldsymbol{\delta}_k^T, \quad \text{if } k = k^*. \tag{38}$$

Each prototype moves along directions determined by its own local metric Λ_k rather than a shared global one. Each matrix is normalised independently after every step:

$$\Omega_k \leftarrow \Omega_k \cdot \sqrt{\frac{m}{\operatorname{tr}(\Omega_k^T \Omega_k)}}, \quad k = 1, \ldots, K. \tag{39}$$

Since only the winner receives gradient signal, every Ω_k covers the local geometry of its Voronoi cell independently of all other prototypes. The resulting isolation is particularly useful in OCC. A prototype covering a dense core region of the target class learns a tight, discriminative metric, whereas a peripheral prototype retains a wider acceptance volume that accommodates sparser target samples near the boundary.

3.7 OC-GTLVQ: Tangent Subspace Distances

The preceding variants all employ additive metrics that amplify distances along discriminative directions. OC-GTLVQ takes the opposite approach, as it identifies directions of tolerated variation and removes them from the distance computation. The formulation follows the tangent distance paradigm of Saralajew and Villmann [10]. Each prototype $\mathbf{w}_k$ is associated with an orthonormal basis $\Psi_k \in \mathbb{R}^{m \times p}$ that satisfies $\Psi_k^T \Psi_k = I_p$. Here, $p \ll m$ denotes the subspace dimension. The tangent distance measures the norm of the displacement projected onto the orthogonal complement of the tangent subspace:

$$d_{\Psi_k}(\mathbf{s}, \mathbf{w}_k) = \|(I - \Psi_k \Psi_k^T)\boldsymbol{\delta}_k\|^2 = \|P_k \boldsymbol{\delta}_k\|^2, \tag{40}$$

where $P_k = I - \Psi_k \Psi_k^T$ is the orthogonal projector and $\boldsymbol{\delta}_k = \mathbf{s} - \mathbf{w}_k$. The resulting cost function is

$$E_{\text{OC-GTLVQ}}(\mathbf{W}, \boldsymbol{\alpha}, \{\Psi_k\}) = \frac{1}{N} \sum_{\mathbf{s} \in \mathbf{S}} f_\beta(\mu^{\Psi_k}(\mathbf{s}) + \gamma), \tag{41}$$

with μ^{Ψ_k} evaluated under the tangent distance (40). OC-GTLVQ is particularly significant to one-class settings where the target distribution lies near a low-dimensional manifold embedded in high-dimensional space. Expressing the distance as $d_{\Psi_k} = \|\boldsymbol{\delta}_k\|^2 - \|\Psi_k^T \boldsymbol{\delta}_k\|^2$ and writing $\mathbf{u}_k = \Psi_k^T \boldsymbol{\delta}_k \in \mathbb{R}^p$, the distance gradients are given by

$$\frac{\partial d_{\Psi_k}}{\partial \mathbf{w}_k} = -2\,P_k\,\boldsymbol{\delta}_k, \qquad \frac{\partial d_{\Psi_k}}{\partial \Psi_k} = -2\,\boldsymbol{\delta}_k\,\mathbf{u}_k^T. \tag{42}$$

Substituting into the general gradient framework (8) results in the prototype and tangent basis gradients:

$$\frac{\partial E}{\partial \mathbf{w}_k} = f'_\beta \cdot \xi_k^+ \cdot (-2\,P_k\,\boldsymbol{\delta}_k), \tag{43}$$

$$\frac{\partial E}{\partial \Psi_k} = f'_\beta \cdot \xi_k^+ \cdot (-2\,\boldsymbol{\delta}_k\,\mathbf{u}_k^T). \tag{44}$$

The resulting update rules are

$$\mathbf{w}_k \leftarrow \mathbf{w}_k - \eta \cdot f'_\beta \cdot \xi_k^+ \cdot (-2\,P_k\,\boldsymbol{\delta}_k), \tag{45}$$

$$\Psi_k \leftarrow \Psi_k - \eta \cdot f'_\beta \cdot \xi_k^+ \cdot (-2\,\boldsymbol{\delta}_k\,\mathbf{u}_k^T). \tag{46}$$

The prototype update in (45) shifts $\mathbf{w}_k$ strictly along directions orthogonal to the learned tangent subspace, focusing model adaptation on the components that influence distance. The tangent basis update in (46) takes the form of a rank-one outer product between the displacement $\boldsymbol{\delta}_k$ and its projection onto the tangent space $\mathbf{u}_k$, realigning the invariance subspace with the current displacement. When $\boldsymbol{\delta}_k$ is incorporated into the subspace, the tangent distance for the current sample decreases. As a result, the model is moved to treat the displacement as non-discriminative. After every update, Ψ_k undergoes re-orthonormalisation using thin QR decomposition. The requirement that the diagonal elements of R remain positive guarantees a unique factorisation and stable orientation of the basis vectors across all the learning iterations.

3.8 Prediction

At test time, for a query $\mathbf{s}$, the model computes the discriminant (6) without the signed label (setting $\xi = +1$):

$$\hat{\mu}(\mathbf{s}) = \frac{d(\mathbf{s}, \mathbf{w}_{k^*}) - \theta_{k^*}}{d(\mathbf{s}, \mathbf{w}_{k^*}) + \theta_{k^*}}. \tag{47}$$

The decision function maps (47) to a confidence score $g(\mathbf{s}) = 1 - f_\beta(\hat{\mu}(\mathbf{s}))$. It must be noted that values $g(\mathbf{s}) > 0.5$ indicate target-class membership ($d < \theta$) and values $g(\mathbf{s}) < 0.5$ indicate outlier status ($d > \theta$). The threshold at $g = 0.5$ corresponds exactly to the boundary $d(\mathbf{s}, \mathbf{w}_{k^*}) = \theta_{k^*}$.

4 Experimental Setup

Algorithm 1 Metric-Adaptive OC-GLVQ Training

Require: Training data $\mathbf{S}$ with labels, number of prototypes K, metric variant, max iterations T
1: Initialise prototypes $\mathbf{W}$ by selecting K samples uniformly from the target set $\mathbf{S}_\oplus$
2: Initialise α_k via (18) for each prototype
3: Initialise metric parameters ($\boldsymbol{\rho}$, Ω, Ω_k, Ψ_k)
4: **for** $t = 1$ to T **do**
5: Compute distances $d(\mathbf{s}_i, \mathbf{w}_k)$ under current metric for all i, k
6: Determine nearest prototype $k^*(\mathbf{s}_i)$ for each sample
7: Recover thresholds: $\theta_k = \alpha_k^2$
8: Compute discriminants $\mu(\mathbf{s}_i)$ via (6)
9: Compute cost E via (7)
10: Update all parameters via gradient descent on E
11: Apply post-update constraints (trace normalisation, orthogonalisation)
12: **if** early stopping criterion met **then**
13: Restore best parameters and terminate
14: **end if**
15: **end for**

Algorithm 1 summarises the training procedure. The evaluation uses four datasets spanning dimensionalities from 33 to 168, summarised in Table 1. Three of the four datasets are drawn from the UCI Machine Learning Repository [23].

Table 1: Dataset characteristics. $N_\oplus$ and $N_\ominus$ denote target and non-target sample counts.

Dataset	m	N	$N_\oplus$	$N_\ominus$
Steel Plates Fault	33	1941	1268	673
HighDimSynthetic	50	600	297	303
Sonar	60	208	111	97
Musk (clean1)	168	476	269	207

The *Steel Plates Fault* dataset consists of surface defect measurements from steel manufacturing and includes 33 continuous features. *HighDimSynthetic* is generated with scikit-learn's `make_classification`, comprising 10 informative features, 10 redundant features and 30 noise features. These datasets create a controlled environment for evaluating how effectively metric adaptation suppresses noise dimensions. *Sonar* provides 60 frequency-band energy measurements from sonar returns. *Musk (clean1)* includes 168 descriptors of molecular conformation. For each dataset, the majority class serves as the target class $\oplus$ and the minority class as non-target $\ominus$. The standard OCC evaluation protocol requires both classes during training, but the model must learn to characterise the target class and reject outliers. The discriminant (6) requires non-target examples during training. The signed label ξ expands the threshold for target samples and contracts it for non-target samples. Without non-target examples, the thresholds would grow unbounded. All experiments use 10-fold stratified CV with the same random seed for reproducibility. Within each fold, features are standardised to zero mean and unit variance based on statistics computed on the training set only. Four metrics are reported: accuracy, precision, recall and F_1 score. F_1 serves as the primary metric for model selection and comparison, as it balances the trade-off between false acceptances and false rejections that is central to OCC performance. All metrics are computed per fold and averaged.

A grid search is performed for all OC-GLVQ model variants to explore different combinations of hyperparameters, including the number of prototypes $K \in \{3, 5, 7, 10\}$, the transfer function $f \in \{\mathrm{identity}, f_\beta\}$ and the optimiser

selected from {Adam, SGD}. The configuration that achieves the highest F_1 score on the test fold is selected. For the OC-GTLVQ model, the subspace dimension $p \in \{2, 5, 10, 15, 20\}$ is also included in the grid search. The learning rate is initialised at $\eta = 0.01$ and decays to zero using a cosine schedule over 500 iterations. Early stopping with a patience of 20 and restoration of the best parameters are used to avoid overfitting. The sigmoid steepness is set to $\beta = 10$, producing a near-binary transfer function that sharpens the gradient around the decision boundary and remains smooth enough for stable optimisation. Preliminary experiments indicate that values of β in the range $[5, 20]$ give similar performance. The margin is set to $\gamma = 0.1$. All metric-adaptive variants use full-rank matrices with $l = m$. After each gradient step, a metric-specific post-update constraint is applied. Trace normalisation (32) is used for OC-GMLVQ. Trace normalisation (39) is used for OC-LGMLVQ. QR orthogonalisation is used for OC-GTLVQ. For SVQ-OCC, the contrastive cost function is used with Gaussian response ($\alpha = 0.9$) and grid search over the same prototype counts, as it achieved the strongest results among the three variants proposed by Staps et al. [5]. For SVDD with RBF kernel, grid search over $\nu \in \{0.01, 0.05, 0.1, 0.2\}$ is performed using the OC-SVM implementation [12], which is equivalent to SVDD under Gaussian kernels [1].

5 Results

The results are organised around three questions: does metric adaptation improve OCC performance, how do the metric-adaptive variants compare with established baselines and does full-rank projection dominate reduced-rank alternatives? Table 2 presents the F_1 scores across all datasets and models. Table 3 reports the full metric breakdown including accuracy, precision and recall.

Table 2: F_1 score (mean over 10-fold stratified CV). Bold indicates the best result per dataset. Gray shading marks the overall best model. Models are grouped by metric type with baselines below.

Model	Steel	Synthetic	Sonar	Musk
OC-GLVQ	0.838	0.769	0.737	0.722
OC-GRLVQ	0.950	0.921	0.765	0.858
OC-GMLVQ	**1.000**	0.948	0.870	0.947
OC-LGMLVQ	0.999	**0.958**	0.882	**0.963**
OC-GTLVQ	0.994	0.931	**0.888**	0.820
SVQ-OCC	0.780	0.663	0.688	0.706
SVDD	0.788	0.671	0.648	0.649

Across all four datasets, metric-adaptive variants outperform the baseline OC-GLVQ by wide margins. The improvement is progressive with metric complexity: OC-GRLVQ (diagonal) improves over OC-GLVQ by 3–15 percentage points in F_1, OC-GMLVQ (global matrix) adds another 3–11 points and OC-LGMLVQ (local matrices) achieves the highest scores on two of four datasets. The progression corresponds to the supervised case [9, 20] where increasing metric expressiveness yields monotonic gains and extends it to the one-class setting where only a single class boundary has to be characterised. On Steel Plates Fault, OC-GMLVQ achieves a perfect $F_1 = 1.000$, marginally outperforming OC-LGMLVQ (0.999). Hence, the 33-dimensional industrial data with 1941 samples is well served by a single global metric independent of the increased complexity of local projections.

OC-GTLVQ (tangent subspace) outperforms OC-GLVQ on all datasets and OC-GRLVQ on Steel Plates Fault, HighDimSynthetic and Sonar. Allowing the subspace dimension p to be tuned over the range $\{2, 5, 10, 15, 20\}$ pushed OC-GTLVQ to $F_1 = 0.994$ on Steel Plates Fault, close to the full matrix variants. Moreover, OC-GTLVQ attains an F_1 score of 0.888, which outperforms both OC-GMLVQ (0.870) and OC-LGMLVQ (0.882) on the Sonar dataset. The required subspace dimension varies with respect to the datasets, as observed with $p = 20$ dominating on Steel Plates Fault (10/10 folds) while $p = 15$ and $p = 20$ share dominance on HighDimSynthetic. The variation confirms that the optimal tangent subspace dimensionality depends on the intrinsic structure of the data. On Musk and HighDimSynthetic, OC-GTLVQ remains below the full matrix variants, which is consistent with the design of tangent distances [10]. The tangent basis captures a limited number of invariance directions and the remaining orthogonal dimensions jointly encode the target boundary, which is less expressive than a full metric tensor.

The baseline methods (SVQ-OCC [5] and SVDD [1]) perform notably worse than the metric-adaptive OC-GLVQ variants on all the datasets. The performance gap widens with dimensionality. On Musk ($m = 168$), OC-LGMLVQ achieves $F_1 = 0.963$ compared to 0.706 for SVQ-OCC and 0.649 for SVDD, an improvement of 26 and 31 percentage points respectively. The performance gap is consistent across all four metrics reported in Table 3. SVQ-OCC and SVDD do not adapt the distance function to the structure of the input space (Section 2). Hence, both models become uninformative due to concentration of measure effects when a low-dimensional target distribution occupies a high-

Table 3: Full evaluation metrics (mean over 10-fold stratified CV). Bold indicates the best result per dataset and metric. Gray shading marks the best model per dataset.

	Accuracy	Precision	Recall	F_1
Steel Plates Fault ($m = 33$, $N = 1941$)				
OC-GLVQ	0.785	0.855	0.840	0.838
OC-GRLVQ	0.936	0.973	0.931	0.950
OC-GMLVQ	**0.999**	**1.000**	**0.999**	**1.000**
OC-LGMLVQ	**0.999**	0.999	**0.999**	0.999
OC-GTLVQ	0.992	0.995	0.994	0.994
SVQ-OCC	0.643	0.652	0.970	0.780
SVDD	0.665	0.672	0.952	0.788
HighDimSynthetic ($m = 50$, $N = 600$)				
OC-GLVQ	0.802	0.903	0.677	0.769
OC-GRLVQ	0.925	0.952	0.896	0.921
OC-GMLVQ	0.950	0.966	0.933	0.948
OC-LGMLVQ	**0.960**	**0.979**	0.939	**0.958**
OC-GTLVQ	0.932	0.929	0.936	0.931
SVQ-OCC	0.498	0.497	**0.997**	0.663
SVDD	0.652	0.627	0.727	0.671
Sonar ($m = 60$, $N = 208$)				
OC-GLVQ	0.731	0.789	0.712	0.737
OC-GRLVQ	0.760	0.810	0.738	0.765
OC-GMLVQ	0.870	0.919	0.836	0.870
OC-LGMLVQ	**0.885**	**0.953**	0.828	0.882
OC-GTLVQ	0.875	0.854	0.936	**0.888**
SVQ-OCC	0.529	0.532	**0.973**	0.688
SVDD	0.655	0.731	0.605	0.648
Musk (clean1) ($m = 168$, $N = 476$)				
OC-GLVQ	0.641	0.649	0.829	0.722
OC-GRLVQ	0.818	0.806	0.926	0.858
OC-GMLVQ	0.941	0.969	0.926	0.947
OC-LGMLVQ	**0.960**	**0.992**	0.937	**0.963**
OC-GTLVQ	0.757	0.711	**0.970**	0.820
SVQ-OCC	0.548	0.558	0.963	0.706
SVDD	0.544	0.575	0.747	0.649

dimensional attribute space [21]. Metric adaptation overcomes the limitation by learning a projection that amplifies distances along discriminative directions and suppresses irrelevant variation. We observe from the lowest-dimensional dataset (Steel Plates Fault, $m = 33$) that OC-GMLVQ outperforms SVQ-OCC by 22 percentage points in F_1, indicating that the benefit of metric adaptation is not limited to high-dimensional scenarios but extends to moderately-dimensional data where attribute correlations are present. Additional experiments compare full-rank ($l = m$) matrices against reduced dimensions $l \in \{2, 5, 10, 15, 20, 30, 50\}$ for OC-GMLVQ and OC-LGMLVQ, following the limited-rank matrix learning framework of Bunte et al. [22].

Table 4: Effect of projection rank l on F_1 score. Format: OC-GMLVQ / OC-LGMLVQ. Full rank denotes $l = m$.

Rank	Steel ($m=33$)	Synthetic ($m=50$)	Sonar ($m=60$)	Musk ($m=168$)
Full rank	1.000 / 0.999	0.948 / 0.958	0.870 / 0.882	0.947 / 0.963
$l = 20$	1.000 / 1.000	0.940 / 0.921	0.775 / 0.683	0.980 / 0.962
$l = 10$	0.998 / 0.998	0.919 / 0.891	0.646 / 0.484	0.960 / 0.931
$l = 5$	1.000 / 0.999	0.862 / 0.760	0.366 / 0.253	0.201 / 0.241
$l = 2$	0.964 / 0.997	0.564 / 0.480	0.288 / 0.252	0.232 / 0.278

Aggressive rank reduction leads to performance decline on datasets with $m \geq 50$. Limiting the rank for Sonar reduces OC-GMLVQ results from 0.870 to 0.366 when the rank drops from full to $l = 5$. A similar behaviour is seen with the Musk dataset where the scores dropped to 0.201 for OC-GMLVQ and 0.241 for OC-LGMLVQ. The drop indicates that the target boundary occupies a subspace whose critical dimensionality varies across problems. However, Steel

Plates Fault exhibited high performance with remarkable consistency, as is observed with $l = 5$ and also $l = 2$, yielding $F_1 = 0.964$ for OC-GMLVQ. The Steel Plates dataset evidently admits a low-dimensional target description, consistent with its origin in industrial surface inspection where a small number of physical measurements capture the defect structure. A noteworthy exception is where the full-rank result (0.947) is exceeded by OC-GMLVQ at $l = 20$ with $F_1 = 0.980$ for the Musk ($m = 168$) dataset. Additionally observed for Musk is OC-LGMLVQ at $l = 20$ matching full rank (0.962 vs 0.963). In scenarios where the number of matrix parameters ($l \cdot m = 28{,}224$ at full rank) is larger compared to the size of the input training data, a limited rank scheme can be used as an implicit regularisation. Thus, the benefit of using rank-reduction for regularisation purposes is mostly dependent on the dataset under consideration. The evidence from Sonar and HighDimSynthetic indicates that practitioners should validate rank sensitivity on held-out data rather than defaulting to full rank unconditionally.

6 Discussion

The one-class setting amplifies the importance of metric choice relative to multiclass classification. In multiclass LVQ [7, 9], two distance computations (d^+ and d^-) both adapt to class structure and the discriminant normalises their difference. The numerator and denominator are affected by the metric symmetrically, indicating a partial absorption of moderate suboptimality through normalisation. In OCC, only one distance computation exists; the threshold θ_k provides the reference and is itself a learned scalar with no geometric structure. As the results in Section 5 confirm, metric adaptation projects away uninformative directions so that θ_k governs a meaningful boundary in a discriminative subspace, effectively reducing the dimensionality of the problem that the threshold must solve.

The acceptance region together with its form is determined by the interplay between the learned metric and the thresholds. For OC-GRLVQ, the diagonal weights λ_j stretch the boundary into an axis-aligned ellipsoid whose semi-axes scale as $1/\sqrt{\lambda_j}$ whilst the threshold controls the overall size of the ellipsoid. For OC-GMLVQ, θ_k bounds the projected distance $\|\Omega(\mathbf{s} - \mathbf{w}_k)\|^2 < \theta_k$ resulting in an arbitrarily oriented ellipsoid in the actual space whose principal axes align with the singular vectors of Ω. The metric tensor $\Lambda = \Omega^T\Omega$ determines the shape whilst θ_k regulates the volume in a separation that allows efficient joint optimisation. OC-LGMLVQ extends the formulation by equipping each prototype with its own Λ_k, enabling heterogeneous local geometries. For OC-GTLVQ, θ_k bounds the tangent distance $\|P_k(\mathbf{s} - \mathbf{w}_k)\|^2 < \theta_k$, defining a cylindrical region of acceptance with unlimited extent along tangent directions and radius $\sqrt{\theta_k}$ in the orthogonal complement [10]. These results confirm that the metric learning hierarchy extends successfully to the one-class setting. The full-rank matrix variants require $\mathcal{O}(m^2)$ parameters per prototype (for OC-LGMLVQ: $K \cdot m^2$ matrix parameters total). Regarding the high-dimensional data with limited samples, the full-rank parameterisation risks overfitting. The strong results on Musk ($m = 168$, $N = 476$), where the OC-LGMLVQ model obtains $F_1 = 0.963$, indicate that techniques like early stopping and cosine decay serve as adequate regularisation for local matrices. However, in the rank experiment (Table 4), the global matrix at $l = 20$ outperforms full rank on Musk. Therefore, explicit rank reduction serves as effective regularisation when m^2 parameters must be estimated given limited data. The trace normalisation applied after each gradient step (Eq. 32, 39) rescales Ω by a uniform factor, which changes the distance values and can invalidate the second-moment estimates maintained by Adam. In practice, the rescaling factor is close to unity when the learning rate is small, so the perturbation to Adam's state is mild. A similar interaction arises in other projected-gradient settings and does not prevent convergence in the reported experiments, but aggressive learning rates or large gradient steps could increase the effect.

Some prototypes become inactive after training. If a Voronoi cell receives few or no training samples, its associated prototype ceases to receive gradient updates and drifts away from the target distribution. In the reported experiments, all prototypes stayed active across folds when $K \leq 10$. With a higher number of prototypes or in datasets with strong clustering, inactive prototypes are more likely to occur. Monitoring Voronoi cell occupancy during training and removing unused prototypes is therefore recommended.

7 Conclusion

OC-GLVQ and its metric-adaptive extensions OC-GRLVQ, OC-GMLVQ, OC-LGMLVQ and OC-GTLVQ have been presented in this paper. This work establishes a hierarchy of increasingly expressive distance functions for one-class prototype learning, from diagonal feature weighting through global and local matrix adaptation to tangent subspace distances. The cost functions, distance gradients and update rules for all variants have been derived within a unified gradient framework based on the robust GLVQ cost function. A square-root reparameterisation of the threshold parameter has been introduced to ensure stable convergence across all metric variants. The experimental evaluation on real-world datasets indicates consistent and substantial improvements over both SVQ-OCC and SVDD, with per-prototype local projections achieving the strongest overall performance. The effectiveness of metric adaptation for

one-class prototype based learning has been demonstrated. Future work will investigate kernel distances within the OC-GLVQ discriminant, streaming one-class scenarios where the target distribution evolves over time and Neural Gas cooperation to soften the winner-take-all assignment.

References

[1] D. M. J. Tax and R. P. W. Duin. Support vector data description. *Machine Learning*, 54(1):45–66, 2004.

[2] S. S. Khan and M. G. Madden. One-class classification: taxonomy of study and review of techniques. *The Knowledge Engineering Review*, 29(3):345–374, 2014.

[3] P. Perera, P. Oza, and V. M. Patel. One-class classification: a survey. *arXiv preprint arXiv:2101.03064*, 2021.

[4] M. A. F. Pimentel, D. A. Clifton, L. Clifton, and L. Tarassenko. A review of novelty detection. *Signal Processing*, 99:215–249, 2014.

[5] D. Staps, R. Schubert, M. Kaden, A. Lampe, W. Hermann, and T. Villmann. Prototype-based one-class-classification learning using local representations. In *Proc. IEEE Int. Joint Conf. Neural Networks (IJCNN)*, pages 1–8, 2022.

[6] T. Martinetz, S. Berkovich, and K. Schulten. Neural-gas network for vector quantization and its application to time-series prediction. *IEEE Trans. Neural Networks*, 4(4):558–569, 1993.

[7] A. Sato and K. Yamada. Generalized learning vector quantization. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 8, pages 423–429, 1995.

[8] B. Hammer and T. Villmann. Generalized relevance learning vector quantization. *Neural Networks*, 15(8–9):1059–1068, 2002.

[9] P. Schneider, M. Biehl, and B. Hammer. Adaptive relevance matrices in learning vector quantization. *Neural Computation*, 21(12):3532–3561, 2009.

[10] S. Saralajew and T. Villmann. Adaptive tangent distances in generalized learning vector quantization for transformation and distortion invariant classification. In *Proc. Int. Joint Conf. Neural Networks (IJCNN)*, pages 2672–2679, 2016.

[11] D. P. Kingma and J. Ba. Adam: a method for stochastic optimization. In *Proc. Int. Conf. Learning Representations (ICLR)*, 2015.

[12] B. Schölkopf, J. C. Platt, J. Shawe-Taylor, A. J. Smola, and R. C. Williamson. Estimating the support of a high-dimensional distribution. *Neural Computation*, 13(7):1443–1471, 2001.

[13] L. Ruff, R. Vandermeulen, N. Görnitz, L. Deecke, S. A. Siddiqui, A. Binder, E. Müller, and M. Kloft. Deep one-class classification. In *Proc. Int. Conf. Machine Learning (ICML)*, pages 4393–4402, 2018.

[14] T. Villmann, A. Bohnsack, and M. Kaden. Can learning vector quantization be an alternative to SVM and deep learning? *J. Artificial Intelligence and Soft Computing Research*, 7(1):65–81, 2017.

[15] L. Fischer, B. Hammer, and H. Wersing. Efficient rejection strategies for prototype-based classification. *Neurocomputing*, 169:334–342, 2015.

[16] F. Angiulli. Prototype-based domain description for one-class classification. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 34(6):1131–1144, 2012.

[17] T. Kohonen. The self-organizing map. *Proc. IEEE*, 78(9):1464–1480, 1990.

[18] T. Kohonen. *Self-Organizing Maps*. Springer, Berlin, 1995.

[19] T. Villmann, E. Merényi, and B. Hammer. Neural maps in remote sensing image analysis. *Neural Networks*, 16(3–4):389–403, 2003.

[20] M. Biehl, B. Hammer, and T. Villmann. Prototype-based models in machine learning. *WIREs Cognitive Science*, 7(2):92–111, 2016.

[21] K. Q. Weinberger and L. K. Saul. Distance metric learning for large margin nearest neighbor classification. *J. Machine Learning Research*, 10:207–244, 2009.

[22] K. Bunte, P. Schneider, B. Hammer, F.-M. Schleif, T. Villmann, and M. Biehl. Limited rank matrix learning, discriminative dimension reduction and visualization. *Neural Networks*, 26:159–173, 2012.

[23] D. Dua and C. Graff. UCI Machine Learning Repository, 2019. `https://archive.ics.uci.edu/ml`.

A Full Gradient Derivations

The appendix provides step-by-step chain rule derivations that lead to the update rules stated in Section 3. Throughout, $\boldsymbol{\delta}_k = \mathbf{s} - \mathbf{w}_k$ and the abbreviations $d = d(\mathbf{s}, \mathbf{w}_{k^*})$, $\theta = \theta_{k^*}$ for the nearest prototype are used.

A.1 Gradient Factorisation

Every parameter gradient factorises through the chain rule as

$$\frac{\partial E}{\partial \phi} = \underbrace{f'_\beta(\mu+\gamma)}_{\text{sigmoid slope}} \cdot \underbrace{\frac{\partial \mu}{\partial d} \cdot \frac{\partial d}{\partial \phi}}_{\text{distance path}} + \underbrace{f'_\beta(\mu+\gamma)}_{\text{sigmoid slope}} \cdot \underbrace{\frac{\partial \mu}{\partial \theta} \cdot \frac{\partial \theta}{\partial \phi}}_{\text{threshold path}}. \tag{48}$$

For prototype and metric parameters, only the distance path is non-zero ($\partial\theta/\partial\phi = 0$). For the threshold parameter α_k, only the threshold path is non-zero ($\partial d/\partial\alpha_k = 0$). The two discriminant partial derivatives $\xi_k^+ = \partial\mu/\partial d$ and $\xi_k^- = \partial\mu/\partial\theta$ from (9) serve as the coupling factors in every derivation below.

A.2 Reparameterised Threshold (All Variants)

Under $\theta_k = \alpha_k^2$, the chain rule along the threshold path gives:

$$\frac{\partial E}{\partial \alpha_k} = f'_\beta \cdot \frac{\partial \mu}{\partial \theta} \cdot \frac{\partial \theta}{\partial \alpha_k} = f'_\beta \cdot \xi_k^- \cdot 2\alpha_k = f'_\beta \cdot \xi \cdot \frac{-2d}{(d+\theta)^2} \cdot 2\alpha_k. \tag{49}$$

Since $\alpha_k = \sqrt{\theta_k} \sim \mathcal{O}(\sqrt{m})$, the final magnitude is $\mathcal{O}(1/m) \cdot \mathcal{O}(\sqrt{m}) = \mathcal{O}(1/\sqrt{m})$, confirming Theorem 1. The resulting update rule is

$$\alpha_k \leftarrow \alpha_k - \eta \cdot f'_\beta \cdot \xi_k^- \cdot 2\alpha_k. \tag{50}$$

A.3 OC-GLVQ: Euclidean Prototype Gradient

The squared Euclidean distance $d = \|\boldsymbol{\delta}_k\|^2 = (\mathbf{s} - \mathbf{w}_k)^T(\mathbf{s} - \mathbf{w}_k)$ gives the distance gradient

$$\frac{\partial d}{\partial \mathbf{w}_k} = -2\boldsymbol{\delta}_k. \tag{51}$$

Applying the chain rule along the distance path of (48):

$$\frac{\partial E}{\partial \mathbf{w}_k} = f'_\beta \cdot \xi_k^+ \cdot \frac{\partial d}{\partial \mathbf{w}_k} = f'_\beta \cdot \xi \cdot \frac{2\theta}{(d+\theta)^2} \cdot (-2\boldsymbol{\delta}_k) = f'_\beta \cdot \xi \cdot \frac{-4\theta}{(d+\theta)^2}\, \boldsymbol{\delta}_k. \tag{52}$$

The resulting update rule with learning rate $\eta > 0$ is

$$\mathbf{w}_k \leftarrow \mathbf{w}_k - \eta \cdot f'_\beta \cdot \xi_k^+ \cdot (-2\boldsymbol{\delta}_k). \tag{53}$$

A.4 OC-GRLVQ: Relevance and Prototype Gradients

The relevance-weighted distance $d_\lambda = \sum_{j=1}^m \lambda_j\, \delta_{k,j}^2$ depends on both the prototype positions and the relevance weights. Each is derived in turn.

Distance derivatives.

$$\frac{\partial d_\lambda}{\partial w_{k,j}} = -2\lambda_j\, \delta_{k,j}, \tag{54}$$

$$\frac{\partial d_\lambda}{\partial \lambda_j} = \delta_{k,j}^2. \tag{55}$$

Softmax Jacobian. The relevance weights $\lambda_j = \exp(\rho_j)/\sum_l \exp(\rho_l)$ are parameterised through unconstrained ρ_j. The Jacobian entries are:

$$\frac{\partial \lambda_j}{\partial \rho_j} = \lambda_j(1 - \lambda_j), \tag{56}$$

$$\frac{\partial \lambda_l}{\partial \rho_j} = -\lambda_l \lambda_j, \quad l \neq j. \tag{57}$$

Prototype gradient. Substituting (54) into the distance path of (48):

$$\frac{\partial E}{\partial w_{k,j}} = f'_\beta \cdot \xi_k^+ \cdot (-2\lambda_j)\, \delta_{k,j}. \tag{58}$$

Relevance gradient. The chain rule from E to ρ_j passes through all λ_l:

$$\frac{\partial E}{\partial \rho_j} = f'_\beta \cdot \xi_k^+ \cdot \sum_{l=1}^{m} \underbrace{\delta_{k,l}^2}_{=\partial d_\lambda / \partial \lambda_l} \cdot \frac{\partial \lambda_l}{\partial \rho_j}. \tag{59}$$

Expanding with (56)–(57):

$$\frac{\partial E}{\partial \rho_j} = f'_\beta \cdot \xi_k^+ \cdot \lambda_j \left(\delta_{k,j}^2 - \sum_{l=1}^{m} \lambda_l \delta_{k,l}^2 \right) = f'_\beta \cdot \xi_k^+ \cdot \lambda_j \left(\delta_{k,j}^2 - d_\lambda \right). \tag{60}$$

The term $\delta_{k,j}^2 - d_\lambda$ measures how much feature j's squared residual deviates from the current weighted average. Features with above-average residuals receive increased weight; those below average are suppressed. The resulting update rules with learning rate $\eta > 0$ are

$$w_{k,j} \leftarrow w_{k,j} - \eta \cdot f'_\beta \cdot \xi_k^+ \cdot (-2\lambda_j)\, \delta_{k,j}, \tag{61}$$

$$\rho_j \leftarrow \rho_j - \eta \cdot f'_\beta \cdot \xi_k^+ \cdot \lambda_j \left(\delta_{k,j}^2 - d_\lambda \right). \tag{62}$$

A.5 OC-GMLVQ: Matrix and Prototype Gradients

The matrix distance $d_\Omega = \|\Omega \boldsymbol{\delta}_k\|^2 = \boldsymbol{\delta}_k^T \Omega^T \Omega\, \boldsymbol{\delta}_k$ is quadratic in both Ω and $\boldsymbol{\delta}_k$.

Matrix distance derivative. Writing $\mathbf{z}_k = \Omega \boldsymbol{\delta}_k \in \mathbb{R}^l$ gives $d_\Omega = \mathbf{z}_k^T \mathbf{z}_k$. Differentiating with respect to Ω_{ij}:

$$\frac{\partial d_\Omega}{\partial \Omega_{ij}} = 2\, z_{k,i}\, \delta_{k,j}, \tag{63}$$

which in matrix form gives

$$\frac{\partial d_\Omega}{\partial \Omega} = 2\, \mathbf{z}_k\, \boldsymbol{\delta}_k^T = 2\, \Omega\, \boldsymbol{\delta}_k \boldsymbol{\delta}_k^T. \tag{64}$$

Prototype distance derivative. Since $\boldsymbol{\delta}_k = \mathbf{s} - \mathbf{w}_k$, $\partial \boldsymbol{\delta}_k / \partial \mathbf{w}_k = -I$ and

$$\frac{\partial d_\Omega}{\partial \mathbf{w}_k} = -2\, \Omega^T \Omega\, \boldsymbol{\delta}_k = -2\, \Lambda\, \boldsymbol{\delta}_k. \tag{65}$$

Full cost gradients. Substituting into the distance path of (48):

$$\frac{\partial E}{\partial \Omega} = f'_\beta \cdot \xi_k^+ \cdot 2\, \Omega\, \boldsymbol{\delta}_k \boldsymbol{\delta}_k^T, \tag{66}$$

$$\frac{\partial E}{\partial \mathbf{w}_k} = f'_\beta \cdot \xi_k^+ \cdot (-2\, \Lambda\, \boldsymbol{\delta}_k). \tag{67}$$

The resulting update rules with learning rate $\eta > 0$ are

$$\Omega \leftarrow \Omega - \eta \cdot f'_\beta \cdot \xi_k^+ \cdot 2\, \Omega\, \boldsymbol{\delta}_k \boldsymbol{\delta}_k^T, \tag{68}$$

$$\mathbf{w}_k \leftarrow \mathbf{w}_k - \eta \cdot f'_\beta \cdot \xi_k^+ \cdot (-2\, \Lambda\, \boldsymbol{\delta}_k). \tag{69}$$

A.6 OC-LGMLVQ: Local Matrix Gradients

The derivation parallels OC-GMLVQ but each Ω_k belongs exclusively to prototype $\mathbf{w}_k$. The distance gradient for the winning prototype k^* is

$$\frac{\partial d_{k^*}}{\partial \Omega_{k^*}} = 2\, \Omega_{k^*}\, \boldsymbol{\delta}_{k^*} \boldsymbol{\delta}_{k^*}^T. \tag{70}$$

For any non-winner $k \neq k^*$, the cost does not depend on Ω_k for this sample, so $\partial E / \partial \Omega_k = 0$. The full cost gradients are therefore

$$\frac{\partial E}{\partial \Omega_{k^*}} = f'_\beta \cdot \xi_k^+ \cdot 2\,\Omega_{k^*}\,\boldsymbol{\delta}_{k^*}\boldsymbol{\delta}_{k^*}^T, \tag{71}$$

$$\frac{\partial E}{\partial \mathbf{w}_{k^*}} = f'_\beta \cdot \xi_k^+ \cdot (-2\,\Lambda_{k^*}\,\boldsymbol{\delta}_{k^*}), \quad \Lambda_{k^*} = \Omega_{k^*}^T \Omega_{k^*}. \tag{72}$$

The resulting update rules with learning rate $\eta > 0$ are

$$\Omega_{k^*} \leftarrow \Omega_{k^*} - \eta \cdot f'_\beta \cdot \xi_k^+ \cdot 2\,\Omega_{k^*}\,\boldsymbol{\delta}_{k^*}\boldsymbol{\delta}_{k^*}^T, \tag{73}$$

$$\mathbf{w}_{k^*} \leftarrow \mathbf{w}_{k^*} - \eta \cdot f'_\beta \cdot \xi_k^+ \cdot (-2\,\Lambda_{k^*}\,\boldsymbol{\delta}_{k^*}). \tag{74}$$

A.7 OC-GTLVQ: Tangent Basis and Prototype Gradients

The tangent distance $d_{\Psi_k} = \|\boldsymbol{\delta}_k\|^2 - \|\Psi_k^T \boldsymbol{\delta}_k\|^2$ is the difference of two squared norms. Writing $\mathbf{u}_k = \Psi_k^T \boldsymbol{\delta}_k \in \mathbb{R}^p$ and $P_k = I - \Psi_k \Psi_k^T$:

Tangent basis derivative. Differentiating $\|\mathbf{u}_k\|^2 = \boldsymbol{\delta}_k^T \Psi_k \Psi_k^T \boldsymbol{\delta}_k$ with respect to Ψ_k:

$$\frac{\partial \|\mathbf{u}_k\|^2}{\partial \Psi_k} = 2\,\boldsymbol{\delta}_k\,\mathbf{u}_k^T. \tag{75}$$

Since $d_{\Psi_k} = \|\boldsymbol{\delta}_k\|^2 - \|\mathbf{u}_k\|^2$ and the first term does not depend on Ψ_k:

$$\frac{\partial d_{\Psi_k}}{\partial \Psi_k} = -2\,\boldsymbol{\delta}_k\,\mathbf{u}_k^T. \tag{76}$$

Prototype derivative. Both terms depend on $\mathbf{w}_k$ through $\boldsymbol{\delta}_k$. The first gives $\partial \|\boldsymbol{\delta}_k\|^2 / \partial \mathbf{w}_k = -2\boldsymbol{\delta}_k$. For the second, $\partial \|\mathbf{u}_k\|^2 / \partial \mathbf{w}_k = -2\Psi_k \Psi_k^T \boldsymbol{\delta}_k$. Combining:

$$\frac{\partial d_{\Psi_k}}{\partial \mathbf{w}_k} = -2\boldsymbol{\delta}_k + 2\Psi_k \Psi_k^T \boldsymbol{\delta}_k = -2(I - \Psi_k \Psi_k^T)\boldsymbol{\delta}_k = -2\,P_k\,\boldsymbol{\delta}_k. \tag{77}$$

Full cost gradients. Substituting into the distance path of (48):

$$\frac{\partial E}{\partial \Psi_k} = f'_\beta \cdot \xi_k^+ \cdot (-2\,\boldsymbol{\delta}_k\,\mathbf{u}_k^T), \tag{78}$$

$$\frac{\partial E}{\partial \mathbf{w}_k} = f'_\beta \cdot \xi_k^+ \cdot (-2\,P_k\,\boldsymbol{\delta}_k). \tag{79}$$

The resulting update rules with learning rate $\eta > 0$ are

$$\Psi_k \leftarrow \Psi_k - \eta \cdot f'_\beta \cdot \xi_k^+ \cdot (-2\,\boldsymbol{\delta}_k\,\mathbf{u}_k^T), \tag{80}$$

$$\mathbf{w}_k \leftarrow \mathbf{w}_k - \eta \cdot f'_\beta \cdot \xi_k^+ \cdot (-2\,P_k\,\boldsymbol{\delta}_k). \tag{81}$$

A.8 Trace Normalisation

Proposition 2. *The rescaling $\Omega \leftarrow \Omega \cdot \sqrt{m / \operatorname{tr}(\Omega^T \Omega)}$ is the unique orientation-preserving scaling that satisfies $\operatorname{tr}(\Omega^T \Omega) = m$.*

Proof. Let $c = \sqrt{m / \operatorname{tr}(\Omega^T \Omega)}$. Then $\operatorname{tr}((c\Omega)^T (c\Omega)) = c^2 \cdot \operatorname{tr}(\Omega^T \Omega) = m$. Scaling preserves the null space and singular vector directions of Ω, changing only the singular values by the uniform factor c. Uniqueness follows from the fact that any orientation-preserving scaling $\Omega' = c'\Omega$ satisfying the constraint gives $c'^2 = m / \operatorname{tr}(\Omega^T \Omega)$, so $c' = c$ (taking the positive root). □