

---

# Quadric Hypersurface Intersection for Manifold Learning in Feature Space

---

Fedor Pavutnitskiy\*<sup>1</sup>  
Artem Klochkov\*<sup>2</sup>

Sergei O. Ivanov\*<sup>2</sup>  
Viktor Vialov\*<sup>2</sup>

Evgeny Abramov\*<sup>2</sup>  
Anatolii Zaikovskii\*<sup>2</sup>

Viacheslav Borovitskiy\*<sup>2, 4</sup>  
Aleksandr Petiushko<sup>3</sup>

<sup>1</sup>HSE University    <sup>2</sup>St. Petersburg State University    <sup>3</sup>Lomonosov MSU

<sup>4</sup>St. Petersburg Department of Steklov Mathematical Institute of Russian Academy of Sciences

## Abstract

The knowledge that data lies close to a particular submanifold of the ambient Euclidean space may be useful in a number of ways. For instance, one may want to automatically mark any point far away from the submanifold as an outlier or to use the geometry to come up with a better distance metric. Manifold learning problems are often posed in a very high dimension, e.g. for spaces of images or spaces of words. Today, with deep representation learning on the rise in areas such as computer vision and natural language processing, many problems of this kind may be transformed into problems of moderately high dimension, typically of the order of hundreds. Motivated by this, we propose a manifold learning technique suitable for moderately high dimension and large datasets. The manifold is learned from the training data in the form of an intersection of quadric hypersurfaces—simple but expressive objects. At test time, this manifold can be used to introduce a computationally efficient outlier score for arbitrary new data points and to improve a given similarity metric by incorporating the learned geometric structure into it.

## 1 INTRODUCTION

One particularly interesting new area of research for manifold learning is motivated by the recent advances in deep representation learning. In a wide range of

industrial scenarios where deep feature extractor is used as a part of a larger pipeline, a feature space level outlier detector may help tackle the problem of out-of-distribution input data at test time, which, in its turn, may appear due to undertraining, faulty preprocessing or even a deliberate attack. Manifold learning may be used to build such a detector. Moreover, in problems where we need to compare the similarity of different inputs, e.g. in face recognition, geometry-based detector can be used to improve the similarity metric.

Motivated by these problems, we propose a manifold learning technique where the manifold is learned in form of an intersection of quadric hypersurfaces—the zero-sets of quadratic polynomials. Like principal component analysis (PCA), it yields a manifold as a subset of the ambient Euclidean space.

Fitting a quadric hypersurface intersection is posed as an optimization problem of minimizing distances from training dataset to the intersection. Since the geometric distances are computationally expensive to calculate we discuss various approximations. The simplest possible choice gives rise to a close relative of the kernel PCA with quadratic kernel. We are going beyond this basic model and utilize a finer and more robust approximation. Moreover, we introduce new optimization constraints to make the optimization problem equivariant with respect to isometric transformations of the training dataset.

The proposed quadric hypersurface intersection model is much more expressive than the linear one used in PCA, which is the intersection of hyperplanes. It is also more robust, geometry-respecting and scalable than simple variations of kernel PCA. The number of parameters that define the quadric intersection model

---

Proceedings of the 25<sup>th</sup> International Conference on Artificial Intelligence and Statistics (AISTATS) 2022, Valencia, Spain. PMLR: Volume 151. Copyright 2022 by the author(s).

\*Equal contribution. Code available at: [HTTP://GITHUB.COM/SPBU-MATH-CS/QUADRIC-INTERSECTION](http://github.com/SPBU-MATH-CS/QUADRIC-INTERSECTION).  
Correspondence to: [FPAVUTNITSKIY@HSE.RU](mailto:FPAVUTNITSKIY@HSE.RU).

grows quadratically with the dimension, thus making it suitable for moderately high-dimensional spaces, e.g. for feature spaces of deep models. One of the most important features of the proposed technique is that it is amenable to stochastic gradient descent (SGD), which allows (sub)linear scaling with respect to the training dataset size and is straightforward to implement using modern automatic differentiation frameworks.<sup>1</sup>

To showcase the potential of the proposed technique, in Section 5 we consider its application to an industrial level image classification and outlier detection problem.

## 2 SETTING AND RELATED WORK

We aim to propose a manifold learning technique to drive a geometry-based outlier detector which may be used at a feature space level of industrial scale deep representation learners. We are looking at large unlabeled datasets of synthetically structured data and of moderately high dimension (order of hundreds). We need to keep in mind that training data may be contaminated and may possess complex topology, for instance be highly clustered, with unknown or even non-fixed number of clusters (this is common in open set classification problems). Thus, the technique should be expressive, robust and scalable.

The need for a new technique comes from the fact that classical manifold learning algorithms, such as Isomap (Tenenbaum et al., 2000), LLE (Roweis and Saul, 2000), Laplacian eigenmaps (Belkin and Niyogi, 2002), LTSA (Zhang and Zha, 2004) and anomaly detection techniques based on them (e.g. Hein and Maier (2007)) are not readily suitable for large-scale problems without further modifications.

Geometry-motivated anomaly detection techniques like one-class SVM (OCSVM, Schölkopf et al. (2000)), support vector data description (SVDD, Tax and Duin (2004))<sup>2</sup> or the kernel PCA based novelty detector (Hoffmann, 2007) aim to solve similar problems. However, they usually fail to be simultaneously expressive, robust and, most importantly, scalable enough. To prove this point, we evaluate our technique against (the suitable approximations of) these methods in Section 5.

We acknowledge that the idea of describing a point cloud as a zero set of polynomial functions is not novel. As will be explained later, even the simple PCA may be interpreted this way. More recently, Livni et al. (2013) used similar ideas for solving classification problems. However, their singular value decomposition based ap-

<sup>1</sup>An additional justification for the expressiveness of the model can be found in Mumford (2010, Theorem 1).

<sup>2</sup>In most cases this approach is actually equivalent to OCSVM (Lampert, 2009).

proach is not scalable enough for our target setting and thus not really relevant to our further developments. We also acknowledge the related recent papers by Li et al. (2017) and Jung et al. (2012).

## 3 MANIFOLD LEARNING

*Manifold learning*, as a term, refers to a diverse collection of techniques motivated by the *manifold hypothesis* (Fefferman et al., 2016), the statement that natural datasets (e.g. images of pets) lie in the vicinity of a relatively low-dimensional manifold embedded in a higher-dimensional ambient space. Manifold learning is often considered synonymous to *nonlinear dimensionality reduction* (Lee and Verleysen, 2007), though the latter more often refers to data-visualization methods.

There exists a large set of manifold learning techniques, many of them are considered by Ma and Fu (2011). Most of these techniques can be thought of as black boxes which take a point cloud in a high-dimensional Euclidean space and which map every point of the cloud into a point in a low-dimensional Euclidean space. For example, multidimensional scaling algorithms seek the mapping so as to preserve pairwise distances as well as possible, while PCA, viewed through an appropriate lens, tries to preserve most of the data’s variation.

Another natural but less-often studied class of manifold learning techniques tries to characterize the manifold in the vicinity of which the point cloud lies as a submanifold of the ambient Euclidean space. We highlight that this shift in formulation allows one to ask additional questions, such as how far an arbitrary point in the ambient space is from the manifold—a key question for the outlier detection applications. We proceed to discuss this formulation further.

### 3.1 Characterizing Manifolds

We begin by recalling and highlighting a key property of principal component analysis, namely that it characterizes the manifold it finds as a submanifold of the ambient Euclidean space. Given a centered point cloud  $\mathbf{p}_1, \dots, \mathbf{p}_n \in \mathbb{R}^d$ , PCA finds orthonormal vectors  $\mathbf{v}_1, \dots, \mathbf{v}_d \in \mathbb{R}^d$  such that

$$V_k = \{ \mathbf{x} \in \mathbb{R}^d \mid \mathbf{x} = \alpha_1 \mathbf{v}_1 + \dots + \alpha_k \mathbf{v}_k \} \quad (1)$$

is the  $k$ -dimensional linear subspace (thus a submanifold) of the ambient Euclidean space  $\mathbb{R}^d$  that optimally fits the point cloud in a suitable sense. With this definition, it is possible to compute the distance from any point  $\mathbf{p} \in \mathbb{R}^d$  to the closest point of  $V_k$ :

$$d(\mathbf{p}, V_k) = \left\| \mathbf{p} - \sum_{j=1}^k \langle \mathbf{p}, \mathbf{v}_j \rangle \mathbf{v}_j \right\| = \left( \sum_{j=k+1}^d \langle \mathbf{p}, \mathbf{v}_j \rangle^2 \right)^{1/2}, \quad (2)$$

where  $\|\cdot\|$  and  $\langle \cdot, \cdot \rangle$  are standard Euclidean norm and inner product, respectively. In this sense, PCA explicitly characterizes the manifold through (1). Hereinafter we use  $d$  to denote the *geometric distance*: for a subset  $X \subset \mathbb{R}^d$  and a point  $\mathbf{p} \in \mathbb{R}^d$  this distance is given by

$$d(\mathbf{p}, X) = \inf_{\mathbf{x} \in X} \|\mathbf{p} - \mathbf{x}\|. \quad (3)$$

PCA's way of characterizing a manifold is very convenient but relies on the fact that elements of a linear subspace can be represented as linear combinations of a finite collection of basis vectors, which does not directly extend to non-linear domains. However, one can modify this point of view, to make it more amenable to the non-linear setting by considering the linear subspace that PCA finds as the zero set of some vector-valued linear mapping. For a map  $\mathbf{F} : \mathbb{R}^d \rightarrow \mathbb{R}^l$  we set

$$Z(\mathbf{F}) = \{\mathbf{x} \in \mathbb{R}^d \mid \mathbf{F}(\mathbf{x}) = 0\}. \quad (4)$$

Then we have  $V_k = Z(\mathbf{F}^{(k)})$  for  $\mathbf{F}^{(k)} : \mathbb{R}^d \rightarrow \mathbb{R}^{d-k}$  given by  $\mathbf{F}^{(k)}(\mathbf{x}) = (\langle \mathbf{x}, \mathbf{v}_{k+1} \rangle, \dots, \langle \mathbf{x}, \mathbf{v}_d \rangle)$ . This is an instance of a very general way of representing a submanifold, as a zero set of some smooth function: rather than viewing a manifold as the span of a set of basis vectors, we can view a manifold as a solution to the system of equations  $Z(\mathbf{F}) = 0$  for a suitable  $\mathbf{F}$ .

By the preimage theorem (Milnor and Weaver, 1997, §2, Lemma 1), under mild technical assumptions on  $\mathbf{F}$ , we have that  $Z(\mathbf{F})$  is indeed a manifold.

In the following, we will rely on representing a manifold as a zero set of a function, thus shifting the problem of finding a manifold to the problem of finding a function.

Note that this representation is much more expressive (though less explicit) than representing a manifold as an image of  $\mathbb{R}^k$  under some function  $\mathbf{G} : \mathbb{R}^k \rightarrow \mathbb{R}^d$  as for the mapping  $\mathbf{G}(\alpha_1, \dots, \alpha_k) = \alpha_1 \mathbf{v}_1 + \dots + \alpha_k \mathbf{v}_k$  in PCA. Although this method is widely used, for example in autoencoder neural networks, it can only represent manifolds that can be covered by a single chart: this prevents the accurate representation of, for instance, disconnected domains or of the simple sphere or torus.

## 4 INTERSECTIONS OF QUADRICS

As noted in the previous section, the  $k$ -dimensional submanifold of  $\mathbb{R}^d$  that PCA finds from a point cloud  $\mathbf{p}_1, \dots, \mathbf{p}_n \in \mathbb{R}^d$  is a zero set  $Z(\mathbf{F}^{(k)})$  of some linear function  $\mathbf{F}^{(k)} : \mathbb{R}^d \rightarrow \mathbb{R}^{d-k}$ . If we look a little bit deeper at how this manifold is defined, we can see that  $\mathbf{F}^{(k)}$  solves the optimization problem

$$\mathbf{F}^{(k)} = \arg \min \sum_{j=1}^n d(\mathbf{p}_j, Z(\mathbf{F}^{(k)}))^2. \quad (5)$$

Set  $\mathbf{u}_j = \mathbf{v}_{j+k}$  for indices  $1 \leq j \leq d-k$ . Then  $\mathbf{F}^{(k)} = (f_1, \dots, f_{d-k})$  where  $f_j(\mathbf{x}) = \langle \mathbf{x}, \mathbf{u}_j \rangle$  are linear polynomials with coefficients  $\mathbf{u}_j \in \mathbb{R}^d$  such that vectors  $\mathbf{u}_1, \dots, \mathbf{u}_{d-k}$  are orthonormal. It means that (5) has to be optimized over  $d-k$  vectors  $\mathbf{u}_j$  of dimension  $d$  under the constraint that they should form an orthonormal system. Expanding the distance in (5) through (2), we get a simple optimization problem which can either be solved exactly by computing the singular value decomposition (SVD) of a  $d \times n$  matrix, or approximately through gradient-based optimization.

We propose to extend this by considering quadratic polynomials  $f_j$  instead of the linear ones, as components of function  $\mathbf{F} = (f_1, \dots, f_{d-k})$ .

The zero set of a quadratic polynomial (polynomial of degree 2) is called a *quadric hypersurface* or simply a *quadric*. We consider linear polynomials and constants to be special cases of quadratic polynomials and similarly refer to their zero sets as quadrics. The word *hypersurface* is justified by the fact that in the non-degenerate case  $Z(f_j)$  are  $(d-1)$ -dimensional.<sup>3</sup>

Moreover, the zero set  $Z(\mathbf{F})$ , which coincides with the intersection  $Z(\mathbf{F}) = Z(f_1) \cap \dots \cap Z(f_{d-k})$  is usually a  $k$ -dimensional manifold. Intuitively, each of the  $d-k$  quadrics eliminates one dimension of the  $d$ -dimensional ambient space. We do not dwell here on a precise condition for intersection to be a  $k$ -dimensional manifold.

Quadrics in  $\mathbb{R}^2$  are conic sections: ellipses, hyperbolas and parabolas. However already in  $\mathbb{R}^3$  quadrics and their intersections may be much less trivial. This is illustrated on Figure 1. We refer the reader to the paper of Beale et al. (2016) and to the references therein for a brief review of the previous work on the topic of fitting quadrics in low dimension, while proceeding to present an algorithm suitable for high dimension and large datasets.

### 4.1 Fitting a Quadric Intersection

We start with the optimization problem (5), which, at least in theory, is as applicable for intersections of quadrics as it is to linear subspaces (intersections of linear hypersurfaces). Any quadratic polynomial in  $\mathbb{R}^d$  can be written as

$$f(x_1, \dots, x_d) = \sum_{i \leq j} \alpha_{i,j} x_i x_j + \sum_i \alpha'_i x_i + \alpha'', \quad (6)$$

where  $\alpha_{i,j}, \alpha'_i, \alpha'' \in \mathbb{R}$ . We denote the vector of its coefficients by  $\mathbf{v}(f) = (\alpha_{1,1}, \dots, \alpha_{d,d}, \alpha'_1, \dots, \alpha'_d, \alpha'')^\top \in \mathbb{R}^D$ ,

<sup>3</sup>A quadric is non-degenerate if the Hessian matrix of the homogenization  $f^{\text{hom}}(x_1, \dots, x_{d+1}) = x_{d+1}^2 f(x_1/x_{d+1}, \dots, x_d/x_{d+1})$  of the corresponding polynomial  $f$  is non-singular. In this case the quadric is a smooth algebraic variety and thus a manifold of dimension  $d-1$  (Harris, 2013, Example 3.3).

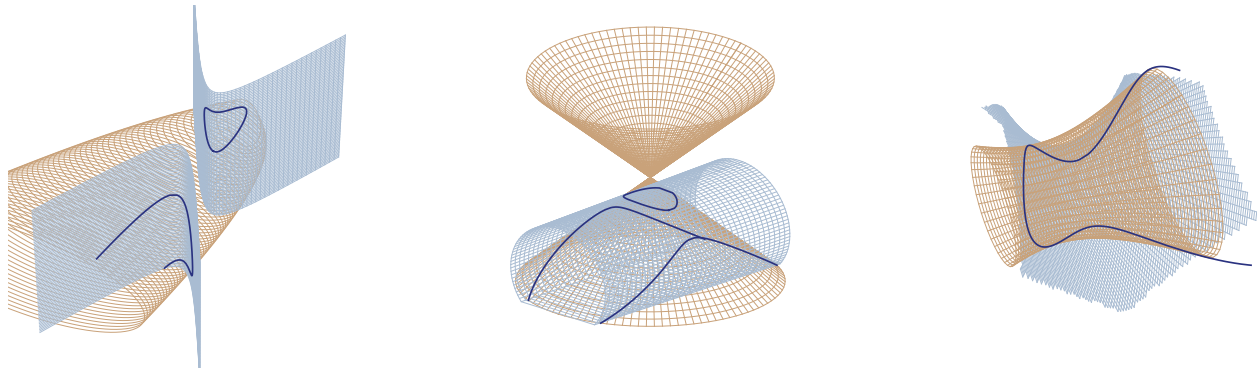


Figure 1: Examples of quadrics and their intersections in  $\mathbb{R}^3$ . Two quadric surfaces are depicted by two meshes, each with its own color. The dark blue line portrays their intersection. From left to right: the intersection of an elliptic paraboloid with a hyperbolic cylinder; the intersection of an elliptic cone with an elliptic cylinder; the intersection of a hyperbolic hyperboloid and a hyperbolic paraboloid.

where  $D = (d^2 + 3d)/2 + 1$  is the number of monomials of degree  $\leq 2$  in  $d$  variables. Thus the intersection of quadrics  $Z(f_1), \dots, Z(f_m)$  is determined by  $m$  vectors  $\mathbf{v}(f_1), \dots, \mathbf{v}(f_m)$ , similarly to how a linear subspace is determined by the basis vectors in PCA.

The intersection of quadrics  $Z(\mathbf{F})$  does not change if we make a replacement  $f_i \mapsto af_i$  for  $a \neq 0$  or  $f_i \mapsto f_i + af_j$  for any  $a \in \mathbb{R}$  and  $i \neq j$ , which is quite clear from viewing the  $Z(\mathbf{F})$  as the solution set of a system of equations. Moreover, the problem (5) per se has  $\mathbf{F}(x) \equiv 0$  for a trivial solution that corresponds to the whole  $Z(\mathbf{F}) = \mathbb{R}^d$ . To handle both of these issues we need to introduce some constraints. We therefore look for an *orthonormal* collection of quadratic polynomials with respect to some inner product, the most simple of which is the standard inner product of vectors  $\mathbf{v}(f_1), \dots, \mathbf{v}(f_m)$ .

It turns out though that optimization problem (5) is very hard to solve for quadrics: computing even a single  $d(\mathbf{p}_j, Z(f_k))$  is a (computationally) hard problem because projecting a point onto a quadric is nontrivial. A number of approximations have therefore been proposed (Taubin, 1991; Taubin, 1993) such as  $d_{\text{alg}}(\mathbf{p}_j, Z(f_k)) := |f_k(\mathbf{p}_j)|$ , called the *algebraic distance*.<sup>4</sup> Approximating  $d(\mathbf{p}_j, Z(\mathbf{F})) \approx (\sum_{k=1}^m |f_k(\mathbf{p}_j)|^2)^{1/2}$ , gives the optimization problem ( $\delta_{kl}$  is the Kronecker delta):

$$(f_1, \dots, f_m) = \arg \min_{\substack{\langle \mathbf{v}(f_k), \mathbf{v}(f_l) \rangle = \delta_{kl} \\ \text{for } k, l = 1, \dots, m}} \sum_{j=1}^n \sum_{k=1}^m |f_k(\mathbf{p}_j)|^2. \quad (7)$$

In Appendix A.1 we show that the problem (7) is deeply related to the polynomial kernel PCA. Despite this connection, we view this problem from a different angle and propose a simple yet fruitful idea of solving (7) by

<sup>4</sup>A number of techniques in the literature may be linked to the algebraic distance, e.g. Coope (1993).

applying stochastic gradient descent to perform *unconstrained* optimization of the corresponding Lagrangian, treating the Lagrange multiplier as a tunable hyperparameter. Building on this idea, we proceed to improve this approach. But first, we discuss its downsides.

**Discussion** The problem (7) has a number of downsides. First, the algebraic distance is a poor approximation of the geometric distance. In practice, this may result in artifacts and unstable behavior of gradient-based optimization. Second, the technique’s deep connection to PCA suggests that it may be very sensitive to outliers, similar to how PCA is (Candès et al., 2011). Finally, the optimization problem (7) does not reflect well the geometric structure of the manifold learning problem as it is non-equivariant in the following sense. If  $f_1, \dots, f_m$  is an orthonormal collection of quadratic polynomials and  $\theta : \mathbb{R}^d \rightarrow \mathbb{R}^d$  is an isometry, then the collection  $f_1 \circ \theta, \dots, f_m \circ \theta$  can fail to solve the optimization problem (7) for the point cloud transformed by  $\theta$  (see Appendix A.3). Below we address these downsides and present a new technique for fitting an intersection of quadrics to a point cloud.

## 4.2 Loss Function

In Taubin (1993), a notion  $d_k(\mathbf{p}, Z(f))$  of *approximation distance of order  $k$*  is defined, building upon the idea of  $k$ -th order Taylor approximation. This distance coincides with the only non-negative root of a certain polynomial  $c_0 + c_1 t + \dots + c_k t^k$  of degree  $k$ , whose coefficients depend on partial derivatives of  $f$  in  $\mathbf{p}$ —full details are given in Appendix A.4. One particularly popular approximation of a distance is the *distance of order 1* given by  $d_1(\mathbf{p}, Z(f)) = |f(\mathbf{p})| / \|\nabla f(\mathbf{p})\|$ . For a quadratic polynomial  $f$ , the  $k$ -distance coincides with the 2-distance for  $k \geq 2$ . Moreover, there is a simple

explicit formula for the 2-distance:

$$d_2(\mathbf{p}, Z(f)) = (\sqrt{h^2 + \|f(\mathbf{p})\| \|f\|_{HS} - h}) / \|f\|_{HS}, \quad (8)$$

where  $h = \|\nabla f(\mathbf{p})\|/2$ , and  $\|f\|_{HS}$  is a certain Hilbert–Schmidt norm defined in (12) below.

The distance of order 2 gives a better approximation of the geometric distance in a number of ways. Firstly, it is, in contrast to the algebraic distance, equivariant. More precisely, for isometries  $\theta : \mathbb{R}^d \rightarrow \mathbb{R}^d$  we have

$$d_2(\theta(\mathbf{p}), Z(f \circ \theta)) = d_2(\mathbf{p}, Z(f)). \quad (9)$$

The proof of this fact can be found in Appendix A.5. Secondly, it is majorized by the geometric distance

$$d_2(\mathbf{p}, Z(f)) \leq d(\mathbf{p}, Z(f)), \quad (10)$$

(see Taubin (1991, §6)) while  $d_{\text{alg}}$  and  $d_1$  are not, the latter may even be infinite. This limits the contribution of outliers to the optimization objective and thus facilitates robustness. Indeed, Equation (10) shows that the 2-distance cannot be arbitrarily large for points that are not geometrically far away from the manifold. This is an advantage over 1-distance: at every point where the gradient  $\nabla f$  vanishes, 1-distance from the quadric  $Z(f)$  to this point will be infinite. Thirdly and lastly, this distance is simple to compute. Hence, for quadrics, the distance of order 2 constitutes the optimal candidate approximation of the geometric distance.

Recall that in order to define a new optimization objective, we have to approximate the distance  $d(\mathbf{p}, Z(\mathbf{F}))$  for  $\mathbf{F} = (f_1, \dots, f_m)$ , not just the distance  $d(\mathbf{p}, Z(f_k))$ . In the original optimization problem (7), we used the  $l^2$ -based term  $(\sum_{k=1}^m |d_{\text{alg}}(\mathbf{p}, Z(f_k))|^2)^{1/2}$  as a proxy for  $d(\mathbf{p}, Z(\mathbf{F}))$ . Here, we suggest to use the  $l^1$ -based term  $\sum_{k=1}^m |d_2(\mathbf{p}, Z(f_k))|$ , based on the consideration that  $l^1$ -loss is more robust to outliers than the squared  $l^2$ -loss. This change also leads to equivariance of the objective, as equivariance of each term of the sum implies equivariance of the whole sum.

### 4.3 Constraints

By replacing  $d_{\text{alg}}$  with  $d_2$ , we have made the optimization objective equivariant with respect to the action of the Euclidean group. Unfortunately, the coefficient-wise inner product for quadratic polynomials is not equivariant and the constrained optimization problem still exhibits geometrically unnatural behavior. To resolve this, we suggest an inner product for quadrics such that the constraint of orthonormality with respect to it makes the whole optimization problem equivariant.

Any quadratic polynomial  $f$  can be represented in form

$$f(\mathbf{x}) = \mathbf{x}^\top \mathbf{A} \mathbf{x} + \mathbf{b} \mathbf{x} + \mathbf{c}, \quad (11)$$

where  $\mathbf{A}$  is a symmetric  $d \times d$  matrix,  $\mathbf{b}$  is a row vector,  $\mathbf{c} \in \mathbb{R}$ . If  $f(x_1, \dots, x_d) = \sum_{i \leq j} \alpha_{i,j} x_i x_j + \sum_i \alpha'_i x_i + \alpha''$ , as in (6) then  $\mathbf{A}_{i,i} = \alpha_{i,i}$  and  $\mathbf{A}_{i,j} = \mathbf{A}_{j,i} = \alpha_{i,j}/2$ ,  $i < j$ .

If  $f$  and  $g$  are quadratic polynomials with corresponding symmetric matrices  $\mathbf{A}$  and  $\mathbf{B}$ , we define their Hilbert–Schmidt (degenerate) inner product as the Hilbert–Schmidt inner product of their matrices

$$\langle f, g \rangle_{HS} = \sum_{i,j} \mathbf{A}_{ij} \mathbf{B}_{ij}, \quad \|f\|_{HS} = \sqrt{\langle f, f \rangle_{HS}}. \quad (12)$$

This inner product is degenerate in the sense that the corresponding norm  $\|f\|_{HS}$  is actually only a seminorm, i.e.  $\|f\|_{HS} = 0$  does not imply  $f = 0$ , this is because it vanishes on polynomials of degree  $\leq 1$ . A collection of quadratic polynomials  $f_1, \dots, f_m$  is called HS-orthonormal if it is orthonormal with respect to the Hilbert–Schmidt inner product. In particular, an HS-orthonormal collection of quadratic polynomials consists only of polynomials of degree 2.<sup>5</sup>

It is easy to check (see Appendix A.6 for details) that this inner product is equivariant with respect to the action of the Euclidean group. Specifically, for any isometry  $\theta : \mathbb{R}^d \rightarrow \mathbb{R}^d$  the following holds:

$$\langle f, g \rangle_{HS} = \langle f \circ \theta, g \circ \theta \rangle_{HS}. \quad (13)$$

If we define the weighted vector of coefficients by  $\tilde{\mathbf{v}}(f) = (\alpha_{1,1}, \alpha_{1,2}/\sqrt{2}, \dots, \alpha_{d-1,d}/\sqrt{2}, \alpha_{d,d})^\top$ , where all coefficients that correspond to the non-diagonal entries of  $A$  are divided by  $\sqrt{2}$ , then, we have  $\langle f, g \rangle_{HS} = \langle \tilde{\mathbf{v}}(f), \tilde{\mathbf{v}}(g) \rangle$ , with the regular Euclidean inner product on the right-hand side.

There is a number of ways to enforce orthonormality of  $\tilde{\mathbf{v}}(f_1), \dots, \tilde{\mathbf{v}}(f_m)$ . The problem is well-studied in the context of orthogonality of filters inside layers of neural networks (Bansal et al., 2018). We propose to use the soft orthogonality regularization term  $\|\tilde{V}(\mathbf{F})^\top \tilde{V}(\mathbf{F}) - I\|_{HS}^2$ , where  $\tilde{V}(\mathbf{F}) = (\tilde{\mathbf{v}}(f_1), \dots, \tilde{\mathbf{v}}(f_m))$  is a  $d^{(d+1)/2} \times m$ -matrix, whose columns are  $\tilde{\mathbf{v}}(f_i)$ .

### 4.4 Summary: an Outlier-robust and Equivariant Algorithm

Assume that we have a cloud of points  $\mathbf{p}_1, \dots, \mathbf{p}_n \in \mathbb{R}^d$  and we want to find  $m$  quadratic polynomials  $f_1, \dots, f_m$  so that this cloud lies close to the intersection of quadrics  $Z(f_1) \cap \dots \cap Z(f_m)$ . Recalling that  $\delta_{kl}$  denotes the Kronecker delta, we formally pose

<sup>5</sup>Note that such a collection can still be used to represent a linear subspace, simply because a linear equation of form  $f(\mathbf{x}) = 0$  may be transformed into the quadratic equation  $f(\mathbf{x})^2 = 0$  with the same solution.

the optimization problem as follows

$$(f_1, \dots, f_m) = \underset{\substack{(f_k, f_l)_{HS} = \delta_{kl} \\ \text{for } k, l=1, \dots, m}}{\arg \min} \sum_{j=1}^n \sum_{k=1}^m |d_2(\mathbf{p}_j, Z(f_k))|, \quad (14)$$

where each of  $m$  quadrics  $f_k$  is represented by the  $D$ -dimensional vector of its coefficients. Since both the optimization objective and the constraints are equivariant, the whole optimization problem is equivariant: if quadrics  $f_1, \dots, f_m$  solve the optimization problem above, then for any isometry  $\theta : \mathbb{R}^d \rightarrow \mathbb{R}^d$  the collection  $f_1 \circ \theta, \dots, f_m \circ \theta$  solves the same optimization problem for the transformed point cloud  $\theta(\mathbf{p}_1), \dots, \theta(\mathbf{p}_n)$ .

Compared to the optimization problem (7), closely related to the polynomial kernel PCA, problem (14) facilitates robustness and is equivariant.<sup>6</sup> This is due to a finer distance approximation and  $l^1$ -averaging in the loss and due to Hilbert–Schmidt equivariant constraints. The latter is shown theoretically in Appendix A.6 and the former is illustrated by a toy example in Section 5.1.

In practice, we employ a soft constraint incorporated into the loss that is given by the Lagrangian<sup>7</sup>

$$\sum_{j=1}^n \sum_{k=1}^m |d_2(\mathbf{p}_j, Z(f_k))| + \lambda \left\| \tilde{\mathbf{V}}(\mathbf{F})^T \tilde{\mathbf{V}}(\mathbf{F}) - \mathbf{I} \right\|_{HS}^2, \quad (15)$$

where  $\tilde{\mathbf{V}}(\mathbf{F})$  is as above and  $\lambda$  is a hyperparameter. We solve this via the stochastic gradient descent over the  $D \times m$ -dimensional set of quadric coefficients. Thanks to this, we have (sub)linear scaling with respect to data size and ease of implementation—another two key features of the approach.

#### 4.5 Out-of-distribution Detection

Assuming a moderately high (order of hundreds) dimensional feature space, we may fit an intersection of quadrics to the feature embeddings of the training data, as described in the previous section. The assumption that the embeddings lie close to the found manifold suggests the distance to manifold as a natural outlier score.

Since it is computationally difficult to evaluate this distance exactly, we suggest using the same approximation that was utilized for training. Specifically, we define the outlier score  $\mathfrak{o}(\mathbf{p})$  of an arbitrary point  $\mathbf{p}$  in the embedding space by

$$\mathfrak{o}(\mathbf{p}) = \frac{1}{m} \sum_{k=1}^m d_2(\mathbf{p}, Z(f_k)), \quad (16)$$

<sup>6</sup>Non-equivariance of the original kernel PCA optimization problem (7) is shown in Appendix A.3.

<sup>7</sup>For an additional discussion on the choice of the regularization see Appendix B.5.

where  $f_1, \dots, f_m$  are the quadratic polynomials that define the found manifold. This average, while easy to compute, may serve as an effective out-of-distribution score. In Section 5.2 we evaluate the performance of the out-of-distribution detector built upon it.<sup>8</sup>

#### 4.6 Similarity Robustification

The most natural way to incorporate the geometric structure of a manifold into the similarity measurement procedure is to use the geodesic distance of the manifold as the new dissimilarity function. Unfortunately though, computing the geodesic distance between a pair of points on the intersection of quadrics is a difficult problem rendering such an approach impractical.

On the other hand, a different approach can be used to improve a given (dis)similarity metric (e.g. Euclidean) using the found geometric structure. Namely, by incorporating the information of the outlierness into the similarity function, we can make classification more robust. For instance, we can modify a similarity function  $s$  by declaring the outliers dissimilar to anything:

$$s_h(\mathbf{x}, \mathbf{y}) = \begin{cases} s(\mathbf{x}, \mathbf{y}), & \max(\mathfrak{o}(\mathbf{x}), \mathfrak{o}(\mathbf{y})) < t, \\ 0, & \max(\mathfrak{o}(\mathbf{x}), \mathfrak{o}(\mathbf{y})) \geq t, \end{cases} \quad (17)$$

where  $t$  is a threshold hyperparameter balancing precision and recall and  $s_h$  is the new robustified similarity. This approach is evaluated in Section 5.2 along with the out-of-distribution detector described above.

Apart from the empirical results from Section 5, the robustification given by (17) is supported by the observation that the similarity between outliers is often abnormally large. This may be seen as a consequence of deep feature extractors’ inability to distinguish the out-of-distribution samples.

## 5 EXPERIMENTS

### 5.1 Toy Example

To illustrate the robustness of the proposed technique we study a toy example. Consider the curve shaped like a seam line of a tennis ball, given by the parametrization

$$\begin{aligned} x &= a \cos t + b \cos 3t, & y &= a \sin t - b \sin 3t, & (18) \\ z &= 2\sqrt{ab} \sin 2t, & & \text{with } a = 0.8, b = 0.2, & (19) \end{aligned}$$

and take a cloud of 100 points generated by adding normally distributed noise to random points on the

<sup>8</sup>Note that this score may alternatively be viewed as the combined score of the ensemble of simple out-of-distribution detectors induced by individual quadrics.

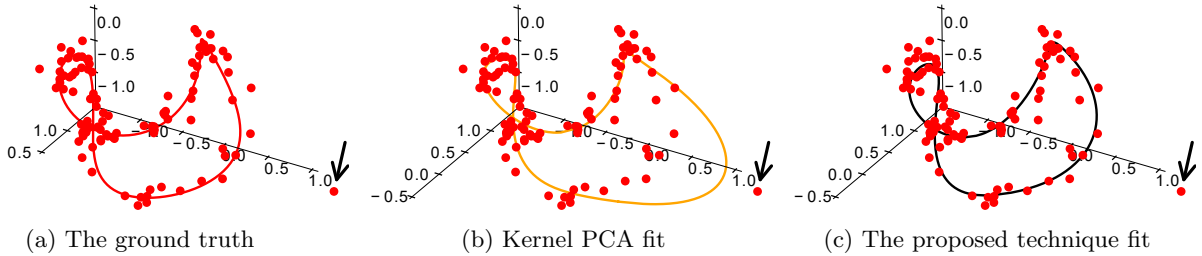


Figure 2: The point cloud of 99 points is generated by adding Gaussian noise to random points lying on the ground truth curve. A random outlier (under the arrow) having twice the norm of the points on the curve is then added to the point cloud. The ground truth (a), the fit corresponding to the exact solution of the optimization problem (7) from Section 4.1 computed by means of kernel PCA with quadratic kernel (b) and the fit of the proposed technique summarized in Section 4.4 (c) are shown.

curve. The curve, a contaminated point cloud and results of application of both the kernel PCA related basic approach from Section 4.1 and the new proposed technique are illustrated on Figure 2. In this Figure the kernel PCA based approach is virtually equivalent (see Appendix A.1 for the details) to minimizing the  $l_2$ -norm of the algebraic distances with Euclidean orthonormality constraints, while the proposed technique utilizes  $l_1$ -norm minimization of the 2-distances with the Hilbert-Schmidt orthonormality constraints.

## 5.2 Outlier Detection for Face Recognition

We consider an image classification pipeline consisting of a face detection and alignment algorithm (MTCNN, Xiang and Zhu (2017)), deep feature extractor (ArcFace, Deng et al. (2019)) and cosine similarity based classifier. The feature extractor used was trained on the variation of the MS1M dataset (Guo et al., 2016).<sup>9</sup> In this setting, we fit a quadric intersection manifold to  $\approx 6 \cdot 10^6$  feature space embeddings of photos from the same dataset. The embeddings lie in the 512-dimensional space. Details of the test datasets construction and respective licences are discussed in Appendix B. We then measure the performance of the quadric intersection based outlier detector and the performance of the cosine similarity improved by penalizing the outliers detected by it as by Section 4.6. The quadric based approach is compared to various geometry-based outlier detectors such as principal component analysis (PCA), kernel principal component analysis for novelty detection (KPCA-ND, Hoffmann (2007)) with RBF kernel and kernel one-class support vector machine (OCSVM, Schölkopf et al. (2000)) with

degree 3 polynomial kernel.<sup>10</sup> In all these approaches we normalize the embeddings as a preprocessing step. Finally, we compare our technique to the approach based on the embedding norm (NORM), that is motivated by recent work of Yu et al. (2020) that shows that in the face recognition domain, the norm of an embedding might carry some information on the image outlierness.

Motivated by a simple ablation study (see details in Appendix B.4), we fit the intersection of 100 quadrics (we refer to this approach as Q-FULL) and a 170-dimensional PCA plane to data. Additionally, to showcase the advantages of the proposed technique over the basic idea from Section 4.1, we fit the intersection of 100 quadrics by applying SGD to solve the kernel PCA related optimization problem (7), we refer to this approach as Q-BASE. Since the naïve kernel PCA scales cubically with respect to data size, to use the RBF-based KPCA-ND we need to resort to approximations. Specifically, we utilize 300 random Fourier features (Rahimi and Recht, 2008) to approximate the implicit feature map. OCSVM does not scale favorably with data size as well, thus we train it on a subsample of size  $8 \cdot 10^4$  of the full training data. To examine our method in the low-data regime we also consider an intersection of 30 quadrics fit to the same subsample of size  $8 \cdot 10^4$  of the data, we refer to this as Q-SUB.

Fitting an intersection of 100 quadrics took us 72 hours on a pair of Quadro RTX 8000 GPUs, using an unoptimized implementation, for both methods Q-BASE and Q-FULL. The resulting (weights of the) quadric intersection models occupy around 50MB each.

**Outlier Detection** We construct a contaminated dataset by mixing the in-distribution photos from one of the special face recognition datasets with the out-of-

<sup>9</sup>We used the MS1M-ArcFace dataset from <https://github.com/deepinsight/insightface/wiki/Dataset-Zoo>. In the sequel, for outliers we use the Anime-Faces dataset from <https://github.com/bchao1/Anime-Face-Dataset>.

<sup>10</sup>In our setting, OCSVM is also equivalent to support vector data description (Lampert, 2009).

Table 1: The AUC-ROC scores for different feature space outlier detectors described in the main text. For the larger datasets (all except CPLFW and CALFW) the order of standard deviation was estimated from 10 random subsamplings. For OCSVM the standard deviation is of order  $10^{-2}$ , while for all other methods it does not exceed a number of order  $10^{-3}$ . Here (a), (b), (c), (d), (e) correspond respectively to Kemelmacher-Shlizerman et al. (2016), Cao et al. (2018), Karras et al. (2019), Zheng and Deng (2018), Zheng et al. (2017). (b) has been taken from <https://github.com/deepinsight/insightface/wiki/Dataset-Zoo>.

Dataset	Q-FULL	PCA	Q-BASE	KPCA-ND	Q-SUB	OCSVM	NORM
MSIM-ArcFace	<b>0.97</b>	0.89	0.95	0.66	0.82	0.71	0.75
MegaFace (a)	<b>0.89</b>	0.81	0.87	0.73	0.76	0.76	0.74
VGGFace2 (b)	<b>0.96</b>	0.88	0.94	0.70	0.81	0.83	0.75
FFHQ (c)	<b>0.93</b>	0.90	0.92	0.71	0.82	0.85	0.72
CPLFW (d)	<b>0.93</b>	0.81	0.91	0.67	0.73	0.82	0.75
CALFW (e)	<b>0.98</b>	0.88	0.95	0.71	0.79	0.84	0.70

distribution photos in the approximate ratio of 99 to 1. The out-of-distribution photos contain 235 manually picked photos from the CPLFW dataset (Zheng and Deng, 2018) where a face cannot be uniquely recognized by a human (e.g. photos of people in hockey helmets, photos with multiple faces) and 235 images from the Anime-Faces dataset<sup>7</sup> aligned by the MTCNN. See the resulting AUC-ROC values for different detectors in Table 1.

**Similarity Robustification** Here we evaluate performance of the similarity-based classifier with robustified similarity function. All robustification methods are based upon the equation (17): the outlier score from the corresponding detector is used for thresholding the similarity between given embeddings. For the test scenario we consider the previously-mentioned Cross-Posed Labeled Faces in the Wild (CPLFW) dataset which is considered particularly challenging due to the presence of the pictures that cannot be recognized even by a human (which we consider as outliers). Performance is measured in terms of the identification rate, which can be understood as the true positive rate of the similarity-based classifier solving an identification problem in the presence (Full IR) or in the absence (IR) of distractors taken from the MegaFace dataset, see details in Appendix B.3.

Both CPLFW and MegaFace are split in two halves. The first half is used for choosing the threshold hyperparameter  $t$  and the other half is used to measure performance (in terms of the identification rate). The

Full IR and IR corresponding to each of the robustification methods is presented in Table 2. Q-FULL, Q-BASE and NORM perform similarly in this experiment, outperforming other methods.

### 5.3 Discussion and Method Limitations

Quadric based techniques behave favorably in both the outlier detection and similarity metric robustification problems, improving on the classic baselines. The norm based approach turns out to be a stronger competitor, which is not surprising given that it is specialized to the setting at hand. Our approach, which is generic, matches its performance in similarity metric robustification, and improves upon its performance in outlier detection. The performance of the Q-SUB approach reveals the limitation of our approach: in the low-data regime quadrics-based model is outperformed by the classical geometry-based outlier detection approaches. This is to be expected though, as the proposed technique is designed for use within the big data domain.

## 6 CONCLUSION

We describe a manifold learning technique based on fitting an intersection of quadrics to a point cloud. To make the problem of fitting a quadric intersection to data tractable, we start from the simplest possible approximate formulation that turns out to be deeply related to the polynomial kernel PCA. Analysing its downsides, we proceed to introduce a number of im-

Table 2: Effect of the various modifications of similarity on the identification rate. Methods names are as in Table 1. False positive rate is fixed to  $10^{-5}$ . The corresponding threshold hyperparameters are given in Appendix B.5.

Metric	Initial	Q-FULL	PCA	Q-BASE	KPCA-ND	Q-SUB	OCSVM	NORM
Full IR	0.61	<b>0.67</b>	0.65	0.63	0.59	0.61	0.64	<b>0.67</b>
IR	0.66	0.74	0.72	0.69	0.64	0.66	0.70	<b>0.75</b>



improvements to the formulation that promote robustness and equivariance. The resulting optimization problem is tractable in moderately high dimension, such as the feature space of a deep representation learning model, is amenable to minibatch training, and thus scales well with respect to point cloud size. The learned quadric intersection can be used to define an outlier score and to improve a given similarity metric. We demonstrate and benchmark the proposed approach empirically on an open set image classification task.

## 6.1 Societal impact

The paper is mainly theoretical, presenting a new manifold learning technique suitable for modern application settings, mainly outlier detection and similarity metric improvement at the feature space level of deep representation learning models. The pipelines based on these models are widespread in computer vision and natural language processing where the scalability of the proposed technique allows it to be used as a drop in solution in a wide range of industrial applications.

As the main application of the approach is in the domain of anomaly detection, technique may be used to increase reliability of existing pipelines. Our experiments have shown that our technique may be used to improve the identification rate within a facial recognition framework. Examples of the negative societal impact of misuse of such frameworks are widely known. However, we stress that the technique's advantages are simplicity and generality, and our choice of the experimental setting should not be regarded as determinative.

## Acknowledgments

FP was supported within the framework of the Basic Research Program at HSE University. SI and AZ were supported by the Ministry of Science and Higher Education of the Russian Federation, agreement № 075-15-2019-1619. VB was supported by the Ministry of Science and Higher Education of the Russian Federation, agreement № 075-15-2019-1620.

## References

- N. Bansal, X. Chen, and Z. Wang. Can We Gain More from Orthogonality Regularizations in Training Deep Networks? In *Advances in Neural Information Processing Systems*, volume 31, 2018. Cited on page 5.
- D. Beale, Y.-L. Yang, N. Campbell, D. Cosker, and P. Hall. Fitting quadrics with a Bayesian prior. *Computational Visual Media*, 2(2):107–117, 2016. Cited on page 3.
- M. Belkin and P. Niyogi. Laplacian Eigenmaps and Spectral Techniques for Embedding and Clustering. In *Advances in Neural Information Processing Systems*, volume 14, 2002. Cited on page 2.
- E. J. Candès, X. Li, Y. Ma, and J. Wright. Robust principal component analysis? *Journal of the ACM*, 58(3), 2011. Cited on page 4.
- Q. Cao, L. Shen, W. Xie, O. M. Parkhi, and A. Zisserman. Vggface2: A dataset for recognising faces across pose and age. In *2018 13th IEEE International Conference on Automatic Face & Gesture Recognition*, 2018. Cited on page 8.
- I. D. Coope. Circle fitting by linear and nonlinear least squares. *Journal of Optimization theory and applications*, 76(2), 1993. Cited on page 4.
- J. Deng, J. Guo, N. Xue, and S. Zafeiriou. ArcFace: Additive Angular Margin Loss for Deep Face Recognition. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019. Cited on page 7.
- C. Fefferman, S. Mitter, and H. Narayanan. Testing the manifold hypothesis. *Journal of the American Mathematical Society*, 29(4), 2016. Cited on page 2.
- Y. Guo, L. Zhang, Y. Hu, X. He, and J. Gao. MS-Celeb-1M: A Dataset and Benchmark for Large-Scale Face Recognition. In *Computer Vision – ECCV*, 2016. Cited on page 7.
- J. Harris. *Algebraic Geometry A First Course*. Springer Science & Business Media, 2013. Cited on page 3.
- M. Hein and M. Maier. Manifold Denoising. In *Advances in Neural Information Processing Systems*, volume 19, 2007. Cited on page 2.
- H. Hoffmann. Kernel PCA for novelty detection. *Pattern recognition*, 40(3), 2007. Cited on pages 2, 7.
- S. Jung, I. L. Dryden, and J. S. Marron. Analysis of principal nested spheres. *Biometrika*, 99(3), 2012. Cited on page 2.
- T. Karras, S. Laine, and T. Aila. A Style-Based Generator Architecture for Generative Adversarial Networks. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019. Cited on page 8.
- I. Kemelmacher-Shlizerman, S. M. Seitz, D. Miller, and E. Brossard. The MegaFace Benchmark: 1 Million Faces for Recognition at Scale. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016. Cited on page 8.

- C. H. Lampert. Kernel Methods in Computer Vision. *Foundations and Trends® in Computer Graphics and Vision*, 4(3), 2009. Cited on pages 2, 7.
- J. A. Lee and M. Verleysen. *Nonlinear Dimensionality Reduction*. Springer Science & Business Media, 2007. Cited on page 2.
- D. Li, M. Mukhopadhyay, and D. B. Dunson. Efficient manifold and subspace approximations with spherelets. *arXiv preprint arXiv:1706.08263*, 2017. Cited on page 2.
- S. Liao, Z. Lei, D. Yi, and S. Z. Li. A benchmark study of large-scale unconstrained face recognition. In *IEEE International Joint Conference on Biometrics*, 2014. Cited on page 14.
- R. Livni, D. Lehari, S. Schein, H. Nachliely, S. Shalev-Shwartz, and A. Globerson. Vanishing Component Analysis. In *Proceedings of the 30th International Conference on Machine Learning*, 2013. Cited on page 2.
- Y. Ma and Y. Fu. *Manifold Learning Theory and Applications*. CRC press, 2011. Cited on page 2.
- J. Milnor and D. W. Weaver. *Topology from the Differentiable Viewpoint*. Princeton University Press, 1997. Cited on page 3.
- D. Mumford. Varieties Defined by Quadratic Equations. In *Questions on Algebraic Varieties*. 2010. Cited on page 2.
- A. Rahimi and B. Recht. Random Features for Large-Scale Kernel Machines. In *Advances in Neural Information Processing Systems*, volume 20, 2008. Cited on page 7.
- S. T. Roweis and L. K. Saul. Nonlinear Dimensionality Reduction by Locally Linear Embedding. *Science*, 290(5500), 2000. Cited on page 2.
- B. Schölkopf, R. C. Williamson, A. Smola, J. Shawe-Taylor, and J. Platt. Support Vector Method for Novelty Detection. In *Advances in Neural Information Processing Systems*, volume 12, 2000. Cited on pages 2, 7.
- G. Taubin. An improved algorithm for algebraic curve and surface fitting. In *1993 (4th) International Conference on Computer Vision*, 1993. Cited on pages 4, 12.
- G. Taubin. Estimation of planar curves, surfaces, and nonplanar space curves defined by implicit equations with applications to edge and range image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(11), 1991. Cited on pages 4, 5.
- D. M. Tax and R. P. Duin. Support vector data description. *Machine learning*, 54, 2004. Cited on page 2.
- J. B. Tenenbaum, V. De Silva, and J. C. Langford. A Global Geometric Framework for Nonlinear Dimensionality Reduction. *Science*, 290(5500), 2000. Cited on page 2.
- R. Vidal, Y. Ma, and S. Sastry. Generalized principal component analysis (GPCA). *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2005. Cited on page 11.
- S. Wold. Cross-validatory estimation of the number of components in factor and principal components models. *Technometrics*, 20(4), 1978. Cited on page 14.
- J. Xiang and G. Zhu. Joint Face Detection and Facial Expression Recognition with MTCNN. In *4th International Conference on Information Science and Control Engineering*, 2017. Cited on page 7.
- C. Yu, X. Zhu, Z. Lei, and S. Z. Li. Out-of-Distribution Detection for Reliable Face Recognition. *IEEE Signal Processing Letters*, 27, 2020. Cited on page 7.
- Z. Zhang and H. Zha. Principal Manifolds and Nonlinear Dimensionality Reduction via Tangent Space Alignment. *SIAM Journal on Scientific Computing*, 26(1), 2004. Cited on page 2.
- T. Zheng and W. Deng. Cross-pose LFW: A database for studying cross-pose face recognition in unconstrained environments. Technical report, 2018. Cited on page 8.
- T. Zheng, W. Deng, and J. Hu. Cross-Age LFW: A Database for Studying Cross-Age Face Recognition in Unconstrained Environments. *arXiv preprint arXiv:1708.08197*, 2017. Cited on page 8.

## A Theory

### A.1 Connection to polynomial kernel PCA

Consider a feature map  $\varphi : \mathbb{R}^d \rightarrow \mathbb{R}^D$  that is given by

$$\varphi(p_1, \dots, p_d) = \left( \underbrace{\dots, p_i p_j, \dots}_{\text{pairwise products}}, \underbrace{p_1, \dots, p_d}_{\text{coordinates}}, \underbrace{1}_{\text{constant}} \right)^\top. \quad (20)$$

For a quadratic polynomial  $f$  we have

$$f(\mathbf{p}) = \langle \varphi(\mathbf{p}), \mathbf{v}(f) \rangle, \quad (21)$$

where  $\mathbf{v}(f)$  is the coefficient vector (see Section 4.1).

Let  $f_1, \dots, f_m$  be quadratic polynomials in  $\mathbb{R}^d$  and  $\mathbf{v}(f_1), \dots, \mathbf{v}(f_m) \in \mathbb{R}^D$  be their coefficient vectors that we assume to be orthonormal. We denote by  $\mathcal{V}_m \subseteq \mathbb{R}^D$  the vector space spanned by these vectors and by  $\mathcal{V}_m^\perp$  its orthogonal completion. A simple computation (see Appendix A.2 for details) yields

$$\sum_{j=1}^n \sum_{k=1}^m |f_k(\mathbf{p}_j)|^2 = \sum_{j=1}^n \mathbf{d}(\varphi(\mathbf{p}_j), \mathcal{V}_m^\perp)^2. \quad (22)$$

It follows that the optimization problem is equivalent to minimization of  $\sum_{j=1}^n \mathbf{d}(\varphi(\mathbf{p}_j), \mathcal{V}_m^\perp)^2$ , the same problem that PCA solves. This means that the technique presented above is equivalent to applying PCA in a feature space defined by (20), when viewed from a different angle. Note that here we do not assume that the point cloud  $\varphi(\mathbf{p}_1), \dots, \varphi(\mathbf{p}_n)$  is centered, so the optimization problem (7) is equivalent to the non-centered version of PCA.

A slight modification of the feature map (20) that is given by

$$\tilde{\varphi}(p_1, \dots, p_d) = (\dots, p_i p_j, \dots, \sqrt{2}p_1, \dots, \sqrt{2}p_d, 1)^\top \quad (23)$$

corresponds to the polynomial kernel  $k(\mathbf{x}, \mathbf{y}) = (\langle \mathbf{x}, \mathbf{y} \rangle + 1)^2$  of degree 2 in the sense that  $k(\mathbf{x}, \mathbf{y}) = \langle \tilde{\varphi}(\mathbf{x}), \tilde{\varphi}(\mathbf{y}) \rangle$ . This shows that the optimization problem is closely connected with kernel PCA with a polynomial kernel.<sup>11</sup>

This suggests an alternative way of solving the optimization problem (7)—by computing the SVD of a  $D \times D$  matrix, although it is usually impractical since  $D$  depends quadratically on  $d$  and the computational complexity of SVD is of order  $O(D^3) = O(d^6)$ .

### A.2 Additional details on the connection to polynomial kernel PCA

For each  $\mathbf{u} \in \mathbb{R}^D$  we denote by  $\text{pr}(\mathbf{u})$  its orthogonal projection onto  $\mathcal{V}_m$ . It is defined by the formula

$$\text{pr}(\mathbf{u}) = \sum_{k=1}^m \langle \mathbf{u}, \mathbf{v}(f_k) \rangle \cdot \mathbf{v}(f_k). \quad (24)$$

Therefore the following holds for the distance  $\mathbf{d}(\mathbf{u}, \mathcal{V}_m^\perp)$  from  $\mathbf{u}$  to  $\mathcal{V}_m^\perp$ :

$$\mathbf{d}(\mathbf{u}, \mathcal{V}_m^\perp)^2 = \|\text{pr}(\mathbf{u})\|^2 = \sum_{k=1}^m \langle \mathbf{u}, \mathbf{v}(f_k) \rangle^2. \quad (25)$$

Using the formulas (25) and (21) we obtain

$$\sum_{j=1}^n \sum_{k=1}^m |f_k(\mathbf{p}_j)|^2 = \sum_{j=1}^n \sum_{k=1}^m \langle \varphi(\mathbf{p}_j), \mathbf{v}(f_k) \rangle^2 = \sum_{j=1}^n \|\text{pr}(\varphi(\mathbf{p}_j))\|^2 = \sum_{j=1}^n \mathbf{d}(\varphi(\mathbf{p}_j), \mathcal{V}_m^\perp)^2, \quad (26)$$

which proves equation (22).

<sup>11</sup>The feature map (20) is also similar to the Veronese map of degree 2 used in Generalized PCA (Vidal et al., 2005, §3.1). The difference is we consider not only quadratics, but all monomials of degree  $\leq 2$ .

### A.3 Non-equivariance of the optimization problem (7)

On the pictures below we show an example of two point clouds on the plane and the corresponding quadrics that solve the optimization problem (7) exactly. The point cloud on Figure 3 (b) can be obtained by shifting the point cloud on Figure 3 (a) by vector  $(10, 0)$ . However, the corresponding quadrics are not the shifts of each other.

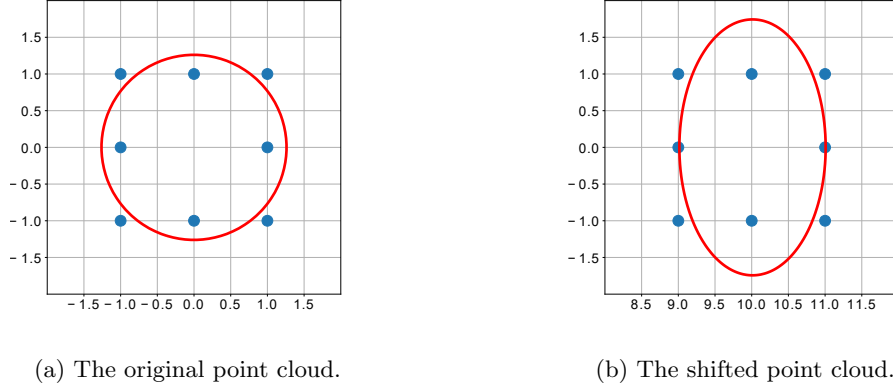


Figure 3: A point cloud and its shifted version (blue) and the respective quadrics (red) that solve the optimization problem (7).

The picture corresponding to the optimization problem from Section 4.4 should be similar to Figure 3 (a) because of the equivariance: the solution of the translated problem is the translated solution of the original problem.

### A.4 Approximation distance of order $k$

Let  $f : \mathbb{R}^d \rightarrow \mathbb{R}$  be a polynomial. For a multi-index  $I = (i_1, \dots, i_d)$  we denote by  $C_I$  the coefficient of the Taylor polynomial of  $f$  in the point  $p$  corresponding to the monomial  $x_1^{i_1} \dots x_d^{i_d}$ :

$$C_I = \frac{1}{I!} \frac{\partial^{|I|} f}{\partial x_1^{i_1} \dots \partial x_d^{i_d}}(p), \quad (27)$$

where  $|I| = i_1 + \dots + i_d$  and  $I! = i_1! i_2! \dots i_d!$ .

For each integer  $l \geq 1$  we set

$$c_l = - \left( \sum_{|I|=l} C_I^2 / b(I) \right)^{1/2}, \quad (28)$$

where  $b(I) = |I|! / I!$  is the multinomial coefficient and  $c_0 = |f(\mathbf{p})|$ . Note that  $c_0 \geq 0$  and  $c_l \leq 0$  for  $l \geq 1$ . With this, the polynomial  $\sum_{l=0}^k c_l t^l$  has a unique non-negative root. We denote this root by  $\mathbf{d}_k(\mathbf{p}, Z(f))$  and, following Taubin (1993), call it *the approximation distance of order  $k$* . Note that  $\mathbf{d}_k(\mathbf{p}, Z(f))$  depends on  $\mathbf{p}$  and  $f$ , not only on  $\mathbf{p}$  and the zero set  $Z(f)$ , despite the notation.

If  $f(x) = \mathbf{x}^\top \mathbf{A} \mathbf{x} + \mathbf{b} \mathbf{x} + \mathbf{c}$  is a quadratic polynomial, then it is easy to check that  $c_2 = \|\mathbf{A}\|_{HS}$ . Using this and the formula for the roots of the general quadratic equation we obtain the formula

$$\mathbf{d}_2(\mathbf{p}, Z(f)) = \frac{\sqrt{h^2 + |f(\mathbf{p})| \|f\|_{HS}} - h}{\|f\|_{HS}}, \quad (29)$$

where  $h = \|\nabla f(\mathbf{p})\|/2$ . In particular, if  $\|f\|_{HS} = 1$ , we have a particularly simple expression

$$\mathbf{d}_2(\mathbf{p}, Z(f)) = \sqrt{h^2 + |f(\mathbf{p})|} - h. \quad (30)$$

### A.5 Equivariance of the approximation distance order $k$

The coefficients of the polynomial  $c_0 + c_1 t + \dots + c_k t^k$  from the definition of  $\mathbf{d}_k(\mathbf{p}, Z(f))$  depend only on the coefficients of the Taylor polynomial of degree  $k$  of the map  $f$  at point  $\mathbf{p}$ . We denote this polynomial by  $T_{k,f,\mathbf{p}}$ . In order to prove the equivariance of the  $k$ -distance, i.e.  $\mathbf{d}_k(\theta(\mathbf{p}), Z(f \circ \theta)) = \mathbf{d}_k(\mathbf{p}, Z(f))$ , it is enough to show the equivariance of the Taylor polynomial, in the sense that

$$T_{k,\mathbf{p},f} \circ \theta = T_{k,f \circ \theta, \theta(\mathbf{p})}. \quad (31)$$

The defining property of the Taylor polynomial is the following:  $T_{k,f,\mathbf{p}}$  is the only polynomial of degree  $\leq k$  such that

$$\|T_{k,f,\mathbf{p}}(\mathbf{x}) - f(\mathbf{x})\| = o(\|\mathbf{x} - \mathbf{p}\|^k) \quad \text{as } \mathbf{x} \rightarrow \mathbf{p}. \quad (32)$$

Any isometry  $\theta : \mathbb{R}^d \rightarrow \mathbb{R}^d$  is of form  $\theta(\mathbf{x}) = \mathbf{Q} \cdot \mathbf{x} + \mathbf{v}$ , where  $\mathbf{Q}$  is an orthogonal matrix and  $\mathbf{v}$  is some vector. Hence  $T_{k,f,\mathbf{p}} \circ \theta$  is also a polynomial of degree  $\leq k$ . The equation (32) then implies

$$\|(T_{k,f,\mathbf{p}} \circ \theta)(\mathbf{x}) - (f \circ \theta)(\mathbf{x})\| = o(\|\theta(\mathbf{x}) - \theta(\mathbf{p})\|^k) = o(\|\mathbf{x} - \mathbf{p}\|^k) \quad \text{as } \mathbf{x} \rightarrow \mathbf{p}. \quad (33)$$

Therefore  $T_{k,\mathbf{p},f} \circ \theta = T_{k,f \circ \theta, \theta(\mathbf{p})}$ .

### A.6 Equivariance of the Hilbert–Schmidt inner product

The Hilbert–Schmidt inner product of matrices can be written as

$$\langle \mathbf{A}, \mathbf{B} \rangle_{HS} = \text{tr}(\mathbf{A}^\top \cdot \mathbf{B}) = \text{tr}(\mathbf{B}^\top \cdot \mathbf{A}), \quad (34)$$

where  $\text{tr}(\cdot)$  denotes the trace of a matrix. It follows that for any orthogonal matrix  $\mathbf{Q}$  the following holds

$$\langle \mathbf{A}, \mathbf{B} \rangle_{HS} = \langle \mathbf{A}\mathbf{Q}, \mathbf{B}\mathbf{Q} \rangle_{HS} = \langle \mathbf{Q}\mathbf{A}, \mathbf{Q}\mathbf{B} \rangle_{HS}. \quad (35)$$

Recall that any isometry  $\theta : \mathbb{R}^d \rightarrow \mathbb{R}^d$  is of form  $\theta(\mathbf{x}) = \mathbf{Q} \cdot \mathbf{x} + \mathbf{v}$ , where  $\mathbf{Q}$  is an orthogonal matrix and  $\mathbf{v}$  is a vector. If  $f$  is a quadratic polynomial with the corresponding matrix  $\mathbf{A}$  (in the sense of equation (11)), then  $f \circ \theta$  is a quadratic polynomial whose corresponding matrix equals to  $\mathbf{Q}^\top \mathbf{A} \mathbf{Q}$ . Therefore the equation (35) implies

$$\langle f \circ \theta, g \circ \theta \rangle_{HS} = \langle \mathbf{Q}^\top \mathbf{A} \mathbf{Q}, \mathbf{Q}^\top \mathbf{B} \mathbf{Q} \rangle_{HS} = \langle \mathbf{A}, \mathbf{B} \rangle_{HS} = \langle f, g \rangle_{HS}. \quad (36)$$

## B Additional experimental details

### B.1 Datasets licenses

1. MS1M-ArcFace was derived from MS1M dataset by InsightFace project, the license of the project applies: <https://github.com/deepinsight/insightface>.
2. Images of MegaFace are licensed under Creative Commons License, details of dataset license are given in <http://megaface.cs.washington.edu/dataset/download.html>.
3. VGGFace2 is licensed under Creative Commons Attribution 4.0 International license, details are given in [https://web.archive.org/web/20171113123726/http://www.robots.ox.ac.uk/~vgg/data/vgg\\_face2/licence.txt](https://web.archive.org/web/20171113123726/http://www.robots.ox.ac.uk/~vgg/data/vgg_face2/licence.txt).
4. FFHQ is licensed under Creative Commons BY-NC-SA 4.0 license by NVIDIA Corporation, details are given in <https://github.com/NVlabs/ffhq-dataset/blob/master/LICENSE.txt>.
5. The licences for CPLFW, CALFW and Anime-Faces datasets are unknown to the authors. The official dataset websites <http://www.whdeng.cn/cplfw/index.html>, <http://whdeng.cn/CALFW/index.html> and <https://github.com/bchao1/Anime-Face-Dataset> do not provide license information.

## B.2 Datasets construction

The embeddings are constructed by the pretrained ArcFace model LResNet100E-IR, ArcFace@ms1m-refine-v2<sup>12</sup>. For MegaFace, FFHQ, CALFW and CPLFW datasets additional alignment was performed by MTCNN. For CPLFW, due to the complex nature of the dataset, the photos where MTCNN failed to detect a face were preserved in the dataset, with no additional preprocessing applied apart from resizing. Aligned versions of MS1M-ArcFace and VGGFace2 datasets were taken from <https://github.com/deepinsight/insightface/wiki/Dataset-Zoo>.

## B.3 Identification rate

Here we describe the procedure for evaluating the performance of the similarity function  $s$  on the identification problem on the set  $X$  of face images with the additional set  $Y$  of distractor images. It is based on the metric called *identification rate* and is widely used in the face recognition domain (Liao et al., 2014).

First, define a family of classifiers  $C_a$ , parameterized by  $a \in \mathbb{R}$  as follows:  $C_a$  declares a pair  $p = (x, y)$  positive (same class), if  $s(x, y) \geq a$ . The corresponding false positive rate and true positive rate are denoted by  $\text{fpr}(a)$  and  $\text{tpr}(a)$  respectively. For a fixed false positive rate  $f \in [0, 1]$  we define the similarity threshold  $\text{sth}(f)$  by

$$\text{sth}(f) = \sup\{a \in \mathbb{R} \mid \text{fpr}(a) \leq f\}. \tag{37}$$

The *identification rate* is defined to be  $\text{tpr}(\text{sth}(f))$ .

Consider the set

$$Q = \{(x, y) \in X \times X \mid \forall d \in Y \quad s(x, y) > \max(s(x, d), s(y, d))\}, \tag{38}$$

which contains pairs *that cannot be distracted* by elements of  $Y$  and consider the binary classifier that declares a pair  $(x, y)$  positive (same class) if it is positive with respect to the classifier  $C_{\text{sth}(f)}$  and it cannot be distracted. The corresponding  $\text{tpr}$  is called the *full identification rate*. We note that another known terminology for IR and Full IR are *verification rate* and *identification rate* respectively. This terminology comes from the corresponding problems in face recognition:

1. The verification problem is to determine from a pair of images whether they are photos of the same person. This problem is typically solved by introducing a similarity measure between images or their embeddings. In a given benchmark like CPLFW, the verification rate is calculated using the pairs suggested by the dataset creators. However, in our experiments we use all possible pairs in CPLFW.
2. The identification problem is to determine the identity of a person on a given image. This problem is typically posed in the setting with distractors.

In our experiments the subset of first  $8 \cdot 10^5$  embeddings of the MegaFace dataset is used as the distractor set  $Y$ .

## B.4 Ablation study

The results of the simple ablation study used to identify the best number of quadrics in intersection and the optimal number of principal components in the setting of Section 5.2 are presented here. AUC-ROC and IR scores are given in Table 3 and Table 4 respectively. The preliminary selection of numbers of principal components were made by studying the eigenvalue decay.

Another way to guess the optimal number of quadrics is to study the eigenvalues of PCA by means of Wold invariant (Wold, 1978) or plotting and analyzing  $-\log \lambda_n / \lambda_1$ . If the estimated dimension of the linear manifold is  $d - k$ , then  $k$  may serve as the number of quadrics or as its initialization for further tuning. The problem-specific limitation on the model size that often exists in practice may also give an upper bound on the number of quadrics.

## B.5 Methods implementation details and hyperparameters

Quadrics intersection fitting was implemented in `torch` framework, the corresponding model and training routines are included in the repository. PCA and one-class SVM methods implementations from `sklearn` library were used.

---

<sup>12</sup><https://github.com/deepinsight/insightface/wiki/Model-Zoo>

Table 3: AUC-ROC scores for the quadrics based outlier detector depending on the number of quadrics in the intersection and AUC-ROC scores for the PCA based outlier detector depending on the number of principal components. Standard deviations do not exceed  $10^{-3}$ .

Dataset	Quadrics			PCA		
	50 quadrics	100 quadrics	200 quadrics	130-dim	170-dim	200-dim
MS1M-ArcFace	0.97	0.97	0.97	0.88	0.89	0.87
MegaFace	0.88	0.89	0.89	0.79	0.81	0.81
VGGFace2	0.95	0.96	0.96	0.86	0.88	0.86
FFHQ	0.92	0.93	0.93	0.89	0.90	0.89
CPLFW	0.92	0.93	0.93	0.80	0.82	0.80
CALFW	0.97	0.98	0.97	0.88	0.88	0.85

Table 4: IR score for the face identification problem with quadric-based and PCA-based robustification depending (respectively) on the number of quadrics in the intersection or on the number of principal components.

Metric	Quadrics			PCA		
	50 quadrics	100 quadrics	200 quadrics	130-dim	170-dim	200-dim
Full IR	0.632	0.635	0.635	0.620	0.626	0.710
IR	0.737	0.741	0.742	0.611	0.720	0.700

For kernel PCA based novelty detector the implementation from <https://github.com/Nmerrillvt/kPCA> was used together with random Fourier features implementation from <https://github.com/tiskw/random-fourier-features>.

The constrains in the optimization problem (14) require optimization over the set of orthonormal frames relation. Because of this, stochastic gradient descent over the Stiefel manifold could be a natural choice for solving the constrained optimization problem. However, in the preliminary experiments we observed that the simple soft regularization (as in Equation (15)) with  $\lambda = 1$ , is more effective and efficient compared to the manifold optimization. The latter does not require tuning  $\lambda$  but may introduce other algorithm-specific hyperparameters. In our experiments the error term  $\|\tilde{V}(\mathbf{F})^T \tilde{V}(\mathbf{F}) - I\|_{HS}^2$  was of order  $10^{-5}$ , which was small enough for our purposes.

The typical values of the threshold hyperparameter  $t$  in the similarity robustification experiment are given in Table 5. They were determined in a series of 10 experiments. In each experiment, both CPLFW and MegaFace were randomly split in two halves. First half was used for determining threshold  $t$  by means of a grid search and the second one was reserved for evaluating the identification rate.

Robustification based on Q-SUB and KPCA-ND methods leads to IR deterioration for all possible values of the threshold hyperparameter. For other methods, optimal values of threshold correspond to marking around one percent of data as outliers. Because of this, to make comparisons fair, we choose thresholds for Q-SUB and KPCA-ND so as to get one percent of outliers, matching the behavior of other methods.

Table 5: Threshold hyperparameter values for various similarity robustification methods.

	Q-FULL	PCA	Q-BASE	KPCA-ND	Q-SUB	OCSVM	NORM
Threshold	0.033	0.529	0.163	31.47	0.452	$2.9 \cdot 10^{-9}$	14.55