ESAIM: M2AN 58 (2024) 335–361
https://doi.org/10.1051/m2an/2023088

ESAIM: Mathematical Modelling and Numerical Analysis
www.esaim-m2an.org

CONVERGENT AUTOENCODER APPROXIMATION OF LOW BENDING AND LOW DISTORTION MANIFOLD EMBEDDINGS

Juliane Braunsmann[1], Marko Rajković[2], Martin Rumpf[2,*] and Benedikt Wirth[1]

Abstract. Autoencoders are widely used in machine learning for dimension reduction of high-dimensional data. The encoder embeds the input data manifold into a lower-dimensional latent space, while the decoder represents the inverse map, providing a parametrization of the data manifold by the manifold in latent space. We propose and analyze a novel regularization for learning the encoder component of an autoencoder: a loss functional that prefers isometric, extrinsically flat embeddings and allows to train the encoder on its own. To perform the training, it is assumed that the local Riemannian distance and the local Riemannian average can be evaluated for pairs of nearby points on the input manifold. The loss functional is computed *via* Monte Carlo integration. Our main theorem identifies a geometric loss functional of the embedding map as the Γ-limit of the sampling-dependent loss functionals. Numerical tests, using image data that encodes different explicitly given data manifolds, show that smooth manifold embeddings into latent space are obtained. Furthermore, due to the promotion of extrinsic flatness, interpolation between not too distant points on the manifold is well approximated by linear interpolation in latent space.

Mathematics Subject Classification. 49J55, 53Z50, 53B12, 53B50, 65D05, 68T09, 68T07, 68T07.

Received December 8, 2022. Accepted November 2, 2023.

1. Introduction

A central machine learning task is to identify high-dimensional objects from given data sets with points in a (suitably generated) latent manifold. Such a representation is frequently defined *via* an *encoder map* of an *autoencoder* acting on input objects and mapping them into a low-dimensional Euclidean *latent space*. The *latent manifold* is then the image of this encoder map. The associated *decoder* maps back points in latent space to points in the input space. The assumption that an observed high-dimensional data set actually forms a low-dimensional manifold – the image of the latent manifold under the decoder map (which is often called *hidden manifold* due to its *a priori* unknown topology and geometry) – is called the *manifold hypothesis*. Encoder and decoder can be trained as deep neural networks *via* the minimization of a functional called *reconstruction loss* which compares the input data with its image under the composition of the encoder and decoder mapping.

Keywords and phrases. Manifold learning, manifold embedding, autoencoder, latent space, Monte Carlo sampling, Γ-convergence.

[1] University of Münster, Orléans-Ring 10, 48149 Münster, Germany.
[2] Institute for Numerical Simulation, University of Bonn, Endenicher Allee 60, 53115 Bonn, Germany.
*Corresponding author: martin.rumpf@ins.uni-bonn.de

This approach is an instance of *deep manifold learning*. One aims for a latent manifold whose geometry is close to the hidden manifold, however, measuring only a reconstruction loss is not enough to accurately recover the geometry present in the data. Different strategies to favor smoothness of the encoder and decoder mapping and thus regularity of the latent manifold have been investigated. Among them are methods which promote sparsity [42], contractive autoencoders [44] which use loss functionals penalizing the norm of the Jacobian of the encoder mapping, denoising autoencoders [51], or variational autoencoders [26] which regularize the latent representation to match a tractable probability distribution.

A major deficit of autoencoders is that they frequently fail to reproduce the statistical distribution of input data in the latent space. To resolve this issue, approaches promoting *low distortion* or *isometric* encoder maps are proposed. In [46], a loss functional measures the difference between distances in the pushforward metric and distances in latent space in order to quantify the lack of isometry. The loss functional from [38] compares Euclidean distances in latent space with geodesic distances on the input manifold. In [25], isometric, *i.e.* length-preserving, encoder maps are used to more accurately pushforward distributions from input to latent space. A loss function based on Shannon-Rate-Distortion is used for this purpose. The loss functional from [3] promotes isometry of the decoder map (by penalizing deviation from a non-orthogonal Jacobian matrix), and that the encoder is a pseudo-inverse of the decoder (by enforcing the Jacobians of decoder and encoder to be transposes of each other). In [39], isometric embeddings in latent space are learned to obtain standardized data coordinates from scientific measurements. To this end, the Jacobian is approximated *via* normally distributed sampling around each data point (so-called *bursts*), and the deviation of the local covariance of bursts from the identity is used to measure the lack of orthogonality of the Jacobian. In [53], these approaches are extended by studying autoencoders which approximately preserve not only distances, but also angles and areas. This is achieved by a loss functional depending on eigenvalues of the pullback metric defined in terms of Jacobians of encoder and decoder. In [28], the spectral method of Laplacian Eigenmaps [6] was combined with a loss which penalizes, in terms of Lipschitz constants, the deviation of the embedding map and its inverse from the identity.

Data manifolds often come with a metric encoding the cost of local variations on the manifold. For sufficiently regular data manifolds, it is shown in [47] how to transfer this metric to the latent manifold, thereby turning it into a *Riemannian* manifold. This in principle allows to compute shortest paths, exponential maps, and parallel transport in latent space, which the decoder can pushforward to the data manifold. Latent spaces of variational autoencoders were first studied as a Riemannian manifold in [15]. A desired property of autoencoders is to enable meaningful interpolation between data points *via* an affine interpolation in the latent space. In [7] an *adversarial regularizer* is proposed to ensure visually realistic interpolations in latent space. To this end, the adversarial regularizer tries to make the decoding of interpolations in latent space indistinguishable from real data points.

In this paper we study embeddings from general input manifolds of high-dimensional data into low-dimensional latent spaces. We propose a loss functional based on randomly sampled point pairs on the input manifold. We assume that the Riemannian distance on the input manifold and the Riemannian average can be computed for each such pair of points. The loss functional consists of two components. The first component measures the distortion of the distance, thus promoting approximately isometric latent space embeddings. The second component penalizes large bending, thus promoting flatness. Altogether, we obtain a low bending and low distortion (LBD) discrete loss functional. It depends on a parameter ϵ, defined as the maximal distance of each point pair sampled on the input manifold. As the number of samples increases, we obtain a Monte Carlo limit of the initially discrete loss functional as a nonlocal loss functional, a double integral over the pairs of input manifold points which are at most ϵ apart. The main theorem of this paper identifies the Γ-limit of these nonlocal Monte Carlo limits, for sampling distance $\epsilon \to 0$, as a local geometric loss functional of embeddings. This loss functional consists of a distortion and a bending energy that are proper limits of their discrete counterparts.

From an abstract numerical analysis point of view, the above limit problem belongs to the class of problems in which an infinite-dimensional optimization problem is approximated in two steps: First, the model or objective functional is approximated introducing an auxiliary length scale parameter ϵ. Then the new objective functional is restricted to a space or set of discretized functions, parametrized by finitely many parameters that can be

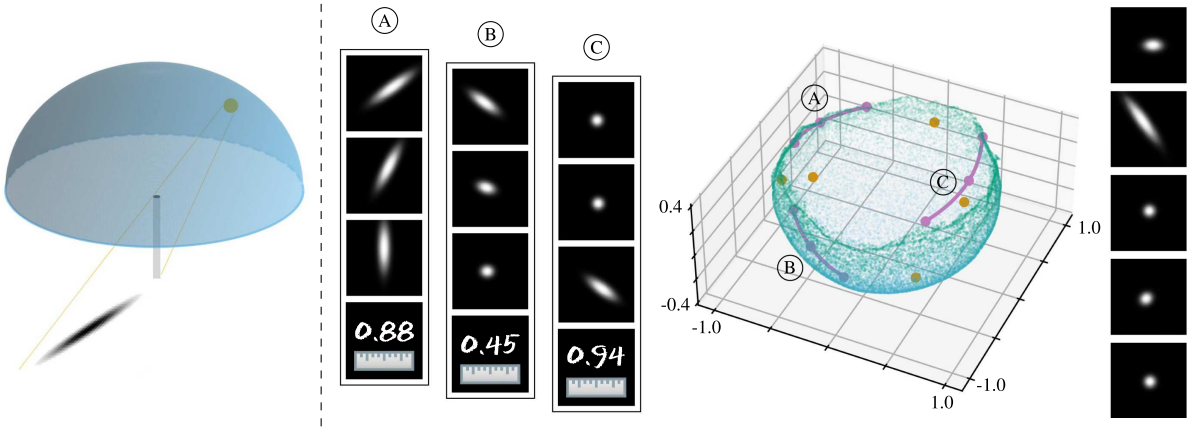

FIGURE 1. Results for dataset (**S**) (*cf.* Sect. 4.1) for $\lambda = 0$ and $\epsilon = \pi/8$ (with maximal distance being π). From left to right: a sketch of the sundial configuration; training data (image pairs with their geodesic average and distance); the latent manifold $\phi(M) \subset \mathbb{R}^{16}$ projected into $\mathbb{R}^3$ *via* PCA; decoder outputs for the orange points in latent space. The encoder was trained separately from the decoder and the training was stopped after the value of the functional $E^{\mathcal{S}_\epsilon}[\phi]$ evaluated on a test set did not decrease for 20 epochs. Pairs with distance below $\frac{1}{100}$ of the maximal distance were rejected. The decoder was trained until an accuracy of 10^{-5} was reached on the same test set. The percentage of explained variance for the first three components lies at 97.8%, with a threshold of 99% reached at 6 components.

optimized for. To relate the final discrete problem to the original one, both the limit $\epsilon \to 0$ and the limit of increasing expressivity of the discrete function space have to be tackled. At the same time, the parameter ϵ and the chosen discrete function space have to be compatible with each other. For instance, in the well-known example of finite element discretizations of phase field approximations to sharp interface problems, the grid size needs to decrease faster than the phase field parameter ϵ in order to avoid discretization artifacts (*cf.* [17]). In our setting, the discrete function space consists of realizations of deep neural networks, and, for our type of loss functional, it becomes necessary that these functions are sufficiently regular with respect to ϵ, *i.e.*, that ϵ decreases faster than some regularity measure of the discretized functions. We achieve this by imposing easily implementable bounds on the network weights which become weaker for decreasing ϵ.

In the presented numerical tests we use digital image data that represents different explicitly given hidden data manifolds. Thus, the dimension of our input space equals the number of pixels times the number of colors.

Figure 1 shows an example with input images representing shadows of a sundial. The learned embedded manifold is illustrated by randomly sampling the input data and visualizing the corresponding output of the trained encoder network. For this visualization, the full latent space was projected onto the three dominant PCA directions of the obtained point cloud in latent space. The input manifold consists of shadow images for all possible positions of a lightsource on a hemisphere. The training data consists of tuples of two such images, their distance and their mean, where the Riemannian distance and mean are induced by the geometry of the hemisphere. The figure demonstrates that our LBD manifold embedding clearly reflects the geometric characteristics of the original hidden manifold, the hemisphere. In contrast to other isometry promoting approaches the *Jacobian of the encoder is not approximated* (*via* backpropagation or complete correlation of all nearest neighbors), but a simple *Monte Carlo sampling of point pairs* is used. Our numerical experiments confirm that the bending loss significantly increases *smoothness* of the resulting latent space manifold when compared to a pure isometry loss. Indeed, solely using an isometry loss functional allows quite irregular Nash–Kuiper embeddings.

Furthermore, the decoder maps linear interpolations in latent space to reasonable interpolations on the data manifold.

The underlying loss functional has been proposed already in the conference paper [14], which contains a short derivation of our embedding ansatz and motivates the loss functional. The only significant theoretical result was a consistency result using Taylor expansion arguments, under the implicit assumption that the manifold embedding is already known to be sufficiently smooth. This paper significantly extends the results in the following directions:

We show existence of minimizers for the nonlocal loss functional in a specific class of functions which consists of realizations of smooth neural networks satisfying (readily enforceable) bounds in certain norms.

We discuss the dependence of the limit functional on the used sampling strategy.

We prove the variational convergence (more specifically, Mosco-convergence as a variant of Γ-convergence) of the nonlocal autoencoder loss functionals to a continuous second order deformation functional. As a consequence, minimizers of the autoencoder loss functionals (which are realizations of neural networks satisfying the above-mentioned norm bounds) converge to a minimizer of the limit functional. This moreover automatically implies the existence of minimizers of the latter.

We present novel and systematic numerical experiments. A substantial part of the result section is devoted to a thorough quantitative analysis of the experiments presented in our conference paper and a detailed study of an additional example with the Klein bottle as the underlying hidden manifold geometry.

The paper is structured as follows. Section 2 introduces the regularization loss as the Monte Carlo limit for dense sampling of input data. In Section 3 we derive the Γ-convergence result for the regularization loss. We start with a recapitulation of some facts about the function spaces necessary in our study in Section 3.1. In Section 3.2,we prove the existence of a minimizer for the nonlocal regularization functional under suitable assumptions on the class of embedding networks. Finally, in Section 3.3 we demonstrate the Mosco-convergence of the discrete functionals to the continuous limit functional. In Section 4 we present some numerical experiments. As a proof of concept, we train autoencoders on several image datasets representing *a priori* known data manifolds. We describe the datasets in Section 4.1 and the autoencoder setup in Section 4.2. Finally, we discuss the numerical results in Sections 4.3 to 4.5.

2. A low bending and low distortion regularization for encoders

We consider a compact, m-dimensional Riemannian manifold $\overline{M}$ with metric g, with or without boundary, embedded in some very high-dimensional space $\mathbb{R}^n$. In our applications $n = N$ for grayscale or $n = 3N$ for RGB images with N pixels. Our aim is to compute an embedding ϕ of $\overline{M}$ into $\mathbb{R}^l$, where we think of the dimension l as being only moderately larger than m (*e.g.* $l = 2m$ so that the existence of a smooth embedding is guaranteed by Whitney's embedding theorem [52]). We call $\mathbb{R}^l$ the *latent space* and $\phi(\overline{M}) \subset \mathbb{R}^l$ the *latent manifold*. Formulated in terms of autoencoders this amounts to learning a pair of maps

$$\phi = \phi_\theta : \overline{M} \to \mathbb{R}^l, \ \psi_\xi : \mathbb{R}^l \to \mathbb{R}^n \text{ with } \psi_\xi(\phi_\theta(x)) \approx x \text{ for all } x \in \overline{M}.$$

The *encoder* ϕ_θ and *decoder* ψ_ξ are implemented as deep neural networks with vectors of encoder and decoder parameters θ and ξ. The encoder ϕ_θ is actually defined on the embedding space $\mathbb{R}^n$, but we don't distinguish between ϕ_θ and $\phi_\theta|_{\overline{M}}$, since we are not interested in ϕ_θ outside of $\overline{M}$. The parameters are typically optimized *via* the minimization of a loss function measuring the difference between the original points x and their reconstructions $\psi_\xi(\phi_\theta(x))$. An appropriate structure and regularity of the embedding into latent space can be promoted by a suitable, geometrically inspired regularization loss for the encoder. This is of particular interest for downstream tasks such as classification [3], Riemannian interpolation and extrapolation [7], clustering or anomaly detection [55]. Indeed, if $\overline{M}$ were isometric to $\mathbb{R}^m$ and thus could be embedded isometrically into an m-dimensional affine subspace of $\mathbb{R}^l$, then, for instance, the most important and basic operations of computing distances and interpolations would become trivial in latent space. The same holds for many other downstream tasks. Although such an embedding is usually prevented by the intrinsic or the global geometry of $\overline{M}$, the hope

for simplified downstream tasks motivates the search for an embedding as close as possible to isometric and flat, at least locally. To this end, we suggest the following two simple objectives for any two not too distant points $x, y \in \overline{M}$:

[**I**sometric] The intrinsic Riemannian distance between x and y in $\overline{M}$ should differ as little as possible from the Euclidean distance between the latent codes $\phi(x)$ and $\phi(y)$.

[**F**lat] The encoder image of a (weighted) average between x and y in $\overline{M}$ should deviate as little as possible from the (weighted) Euclidean average between $\phi(x)$ and $\phi(y)$.

Finding an *isometric embedding*, as extensively pursued in the literature (*cf.* Sect. 1), essentially is equivalent to ensuring the first objective (**I**) for infinitesimally close points $x, y \in \overline{M}$. From the mathematical point of view, though, isometry by itself is insufficient to ensure regular embeddings since the family of isometric embeddings is very large and contains quite irregular elements. In particular, Nash–Kuiper embeddings are in general only Hölder differentiable [30,31,36]. Therefore, we suggest (**I**) and (**F**) as stronger objectives:

- The isometry or low distortion objective (**I**) asks that the *intrinsic* distances between x and y in $\overline{M}$ are approximated by the *extrinsic* distances between $\phi(x), \phi(y) \in \mathbb{R}^l$ in latent space (rather than the intrinsic distances in the latent manifold $\phi(\overline{M})$ with the metric induced by $\mathbb{R}^l$, which would define an isometric embedding).
- The flatness or low bending objective (**F**) enforces some second order low bending regularity or flatness on ϕ by requiring that the *geodesic interpolation* between x and y in $\overline{M}$ is well approximated by extrinsic *linear interpolation* in the latent space $\mathbb{R}^l$.

2.1. A low bending and low distortion loss functional

In what follows we give more details on the manifold $\overline{M}$ and the LBD loss functional.

Let (M_0, g) be a smooth m-dimensional Riemannian manifold without boundary. Let M be an open subset of M_0 so that (M, g) is a smooth m-dimensional Riemannian submanifold of M_0 and let $\overline{M}$ be its closure. Further, let $\overline{M}$ be compact and such that $\overline{M}$ is a (not necessarily smooth) m-dimensional manifold, potentially with boundary, in the following sense: For each $x \in \overline{M}$ there exist a (relatively) open neighborhood U of x in $\overline{M}$ and a homeomorphism $\phi\colon U \to V$ with $V = B_1(0) \coloneqq \{y \in \mathbb{R}^m : |y| < 1\}$ for $x \in M$ and $V = B_1(0) \cap H^m$ for $x \in \overline{M} \setminus M$, where

$$H^m \coloneqq \{(y_1, \dots, y_m) \in \mathbb{R}^m \,:\, y_m \geq 0\}$$

is the closed m-dimensional upper half-space.

Let us denote by $d_M(x, y)$ the Riemannian distance between any two points $x, y \in M$. Further, let TM denote the tangent bundle of M, T_xM the tangent space to M at $x \in M$ and $\exp_x$ the Riemannian exponential map defined on (a subset of) T_xM. As mentioned before, for training of our encoder we will employ the Riemannian mean of point pairs $(x, y) \in M \times M$. By Riemannian mean we mean the midpoint of the unique geodesic connecting x and y in M, if it exists. If $y = \exp_x v$, where $v \in T_xM$ is the initial velocity of this unique shortest geodesic connecting x with y, the Riemannian mean is given by $\operatorname{av}_M(x, y) = \exp_x \frac{v}{2}$. If no such $v \in T_xM$ exists (for instance due to nonunique shortest geodesics), the Riemannian mean is not defined. (Note that one could slightly generalize the definition of the Riemannian mean as the midpoint of shortest connecting curves in $\overline{M}$, which may include curves touching the boundary of M. However, this way there might exist several $y \in M$ with same $\operatorname{av}_M(x, y)$, for which reason we refrain from this generalization.) Therefore we will require some natural conditions on the behavior of the Riemannian exponential map associated with M. To state those, let $V_x \subset T_xM$ denote the (automatically star-shaped) largest open set on which $\exp_x$ is defined. We will assume that there exist constants $r_0 > 0$ and $0 < \kappa < \pi/2$ such that the following holds:

(M1) (injectivity condition) For $x \in M$ let $B_{r_0}^{T_xM}(0) \coloneqq \{v \in T_xM \,:\, g_x(v, v) < r_0^2\}$ denote the open ball in tangent space of radius r_0. Then the Riemannian exponential $\exp_x$ is injective on $V_x \cap B_{r_0}^{T_xM}(0)$ for all $x \in M$.

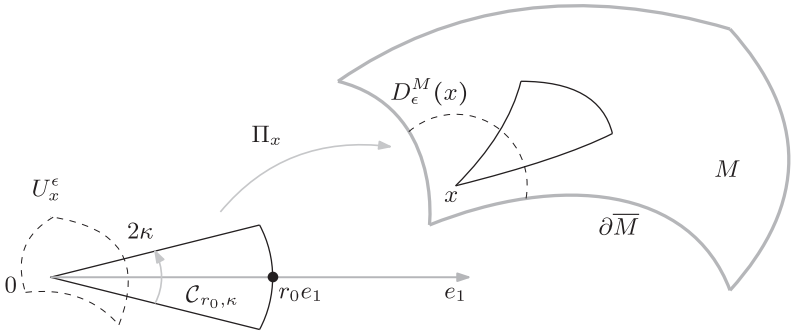

FIGURE 2. The parametrization Π_x that maps the cone $\mathcal{C}_{r_0,\kappa} \subset U_x \subset \mathbb{R}^m$ into M and the neighborhoods $U_x^\epsilon \subset U_x$ onto $D_\epsilon^M(x)$.

(M2′) (strong Lipschitz condition) $\overline{M}$ is strongly Lipschitz, meaning for each $x \in \partial M$ there exists an open neighborhood U of x in M_0, a smooth local chart $h\colon U \to \mathbb{R}^m$ of the manifold M_0 with $h(x) = 0$, $\gamma\colon \mathbb{R}^{m-1} \to \mathbb{R}$ a Lipschitz function with $\gamma(0) = 0$ and $\delta > 0$ such that

$$h(U \cap M) = \{(x', \gamma(x') + t) \,:\, 0 < t < \delta, x' \in \mathbb{R}^{m-1}, |x'| < \delta\}.$$

Note that (M1) is for instance fulfilled when r_0 is chosen as the injectivity radius of the embedding manifold M_0. Both above conditions imply the following cone condition:

(M2) (cone condition) There exists a measurable map $\iota\colon M \times \mathbb{R}^m \to TM$ such that, setting $\iota_x(v) = \iota(x, v)$ for $x \in M, v \in \mathbb{R}^m$, we have that ι_x is an isometric isomorphism between $\mathbb{R}^m$ and T_xM, and V_x contains $\iota_x(\mathcal{C}_{r_0,\kappa})$, where $\mathcal{C}_{r_0,\kappa}$ is the cone

$$\mathcal{C}_{r_0,\kappa} \coloneqq \{w \in \mathbb{R}^m \,:\, 0 \le |w| < r_0,\, |w \cdot e_1| \ge \cos(\kappa)|w|\}$$

of height r_0 and maximum angle κ with the first standard Euclidean basis vector $e_1 \in \mathbb{R}^m$. The dot $\cdot$ denotes the standard inner product in $\mathbb{R}^m$.

The definition of Lipschitz domains in manifolds follows [34]. In the Euclidean setting, it is well known that a Lipschitz domain fulfills the cone condition, see *e.g.* [2]. This similarly holds in our setting, which can be proven in a similar fashion as in [34]. For manifolds without boundary, the largest r_0 coincides with the usual injectivity radius, and we can replace the cone $\mathcal{C}_{r_0,\kappa}$ by the open ball $B_{r_0}^m(0) \subset \mathbb{R}^m$ centered at the origin with radius r_0. For manifolds with boundary, the cone condition is not automatically satisfied: Consider for example the submanifold

$$\overline{M} = \left\{(x, y) \in \mathbb{R}^2 \,:\, y \ge \sqrt{|x|}, y \le 1\right\}$$

of $\mathbb{R}^2$ with the inherited metric. Then no cone can be positioned at $(0, 0)$.

Using the isometric linear mapping $\iota_x\colon \mathbb{R}^m \to T_xM$ from (M2) and setting $U_x \coloneqq \iota_x^{-1}(V_x) \cap B_{r_0}^m(0)$, we can define a local parametrization of M by $\Pi_x\colon U_x \to M$, $\Pi_x \coloneqq \exp_x \circ \iota_x$. This parametrization is known as *normal coordinates around* x (*cf.* Fig. 2). For later notational convenience, we extend Π_x measurably (but arbitrarily) beyond U_x. Let us further introduce the notations $U_x^\epsilon \coloneqq U_x \cap B_\epsilon^m(0)$ and $D_\epsilon^M(x) \coloneqq \Pi_x(U_x^\epsilon)$ for $0 < \epsilon < r_0$. The neighborhood $D_\epsilon^M(x)$ of x consists of all points in the ϵ-neighborhood of x that can be reached from x *via* the Riemannian exponential map (it coincides with the ϵ-neighborhood of x for geodesically convex manifolds M or for points x with Riemannian distance to the boundary larger than ϵ). Thus, condition (M1) ensures that the Riemannian mean of x and any $y \in D_\epsilon^M(x)$ is well-defined.

We can now define the loss functional and the corresponding sampling framework. Let

$$\mathcal{D}_\epsilon \coloneqq \{(x,y) \in M \times M \,:\, y \in D_\epsilon^M(x)\}.$$

For a fixed *sampling radius* $\epsilon \in (0, r_0)$, we consider a finite set $\mathcal{S}_\epsilon \subset \mathcal{D}_\epsilon$ of discrete samples as training data for the encoder. We then define the *discrete sampling loss functional*

$$E^{\mathcal{S}_\epsilon}[\phi] \coloneqq \frac{1}{|\mathcal{S}_\epsilon|} \sum_{(x,y)\in\mathcal{S}_\epsilon} \left(\gamma(|\partial_{(x,y)}\phi|) + \lambda\, |\partial^2_{(x,y)}\phi|^2\right), \tag{2.1}$$

with $\lambda > 0$. The first and second order difference quotients are defined as

$$\partial_{(x,y)}\phi \coloneqq \frac{\phi(y)-\phi(x)}{d_M(x,y)}, \quad \partial^2_{(x,y)}\phi \coloneqq 8\frac{\mathrm{av}_{\mathbb{R}^l}(\phi(x),\phi(y)) - \phi(\mathrm{av}_M(x,y))}{d_M(x,y)^2},$$

where $\mathrm{av}_{\mathbb{R}^l}(a,b) = (a+b)/2$ denotes the linear average in $\mathbb{R}^l$ and $\gamma : [0,\infty) \to [0,\infty)$ a function with a unique minimum $\gamma(1) = 0$. Then, the first term in $E^{\mathcal{S}_\epsilon}$ has a strict minimum for $|\partial_{(x,y)}\phi| = 1$. This promotes $|\phi(x) - \phi(y)| \approx d_M(x,y)$ and thus prefers an approximate isometry with low distortion. An example of such a function, used in the rest of this paper, is

$$\gamma(s) \coloneqq s^2 + \frac{(1+c^2)^2}{s^2+c^2} - 2 - c^2, \quad c > 0.$$

Let us emphasize that $\gamma(s)$ is chosen such that it is finite in $s = 0$ and that the composition $x \mapsto \gamma(\|x\|)$, for $x \in \mathbb{R}^l$, is smooth. The second term in $E^{\mathcal{S}_\epsilon}$ penalizes the deviation of intrinsic averages on $\phi(M)$ from extrinsic ones in $\mathbb{R}^l$. Note that this does not only penalize bending or any extrinsic curvature of $\phi(M)$ in $\mathbb{R}^l$, but in addition it also penalizes deviation of the inplane parametrization of $\phi(M)$ from a linear one. One further notices that the functional is rigid motion invariant by construction, *i.e.*, composition of ϕ with a rigid motion does not change the energy.

2.2. Monte Carlo limit for dense sampling

We next discuss the Monte Carlo limit of the sampling loss functional (2.1) as the number of samples tends to infinity. This limit depends on the sampling strategy. We consider a pair of random variables (X,Y) taking values in $\mathcal{D}_\epsilon$, with a distribution that will be defined below. Let $\mathcal{S}_n$ be a sequence of random finite sets with $\mathcal{S}_n \subset \mathcal{D}_\epsilon$, each containing n independent samples of (X,Y). By the strong law of large numbers [33, Chapter 17.3], we then have almost sure convergence to a *continuous sampling loss functional*

$$\mathcal{E}^\epsilon[\phi] \coloneqq \mathbb{E}\left[\gamma(|\partial_{(X,Y)}\phi|) + \lambda\, |\partial^2_{(X,Y)}\phi|^2\right] = \lim_{n\to\infty} E^{\mathcal{S}_n}[\phi].$$

Let us consider three exemplary ways of sampling, leading to three variants $\mathcal{E}^\epsilon_i$, $i = 1,2,3$ of $\mathcal{E}^\epsilon$:

(S1) Sampling $x \in M$ uniformly and then choosing $y \in D^M_\epsilon(x)$ uniformly. Thus, denoting by V_g the Riemann–Lebesgue volume measure on M (*cf.* [21], Sect. 1.2), we have $\mathbb{P}[X \in A] = V_g(A)/V_g(M)$ and Y is given *via* the conditional law

$$\mathbb{P}[Y \in A | X = x] = \frac{V_g(A \cap D^M_\epsilon(x))}{V_g(D^M_\epsilon(x))}$$

for any measurable $A \subset M$. The limit is

$$\mathcal{E}^\epsilon_1[\phi] \coloneqq \fint_M \fint_{D^M_\epsilon(x)} \gamma(|\partial_{(x,y)}\phi|) + \lambda\, |\partial^2_{(x,y)}\phi|^2 \,\mathrm{d}V_g(y)\,\mathrm{d}V_g(x). \tag{2.2}$$

(S2) Sampling $x \in M$ uniformly and then choosing $v \in U_x^\epsilon$ uniformly and defining $y = \Pi_x(v)$. We have $\mathbb{P}[X \in A] = V_g(A)/V_g(M)$ as above, while Y is given *via* the conditional law as

$$\mathbb{P}[Y \in A | X = x] = \frac{\mathcal{L}(\Pi_x^{-1}(A) \cap U_x^\epsilon)}{\mathcal{L}(U_x^\epsilon)},$$

for any measurable $A \subset M$. Here, $\mathcal{L}$ denotes the m-dimensional Lebesgue measure. This gives

$$\mathcal{E}_2^\epsilon[\phi] := \fint_M \fint_{U_x^\epsilon} \gamma(|\partial_{(x,\Pi_x(v))}\phi|) + \lambda\,|\partial^2_{(x,\Pi_x(v))}\phi|^2 \,\mathrm{d}v\,\mathrm{d}V_g(x), \tag{2.3}$$

where $\mathrm{d}v$ indicates the integration with respect to the Lebesgue measure $\mathcal{L}$.

(S3) Sampling pairs $(x, y) \in \mathcal{D}_\epsilon$ uniformly. We have

$$\mathbb{P}[(X, Y) \in A] = \frac{(V_g \otimes V_g)(A \cap \mathcal{D}_\epsilon)}{(V_g \otimes V_g)(\mathcal{D}_\epsilon)}$$

for any measurable $A \subset M \times M$, so that

$$\mathcal{E}_3^\epsilon[\phi] := \frac{1}{\int_M V_g(D_\epsilon^M(x))\,\mathrm{d}V_g(x)} \int_M \int_{D_\epsilon^M(x)} \gamma(|\partial_{(x,y)}\phi|) + \lambda\,|\partial^2_{(x,y)}\phi|^2 \,\mathrm{d}V_g(y)\,\mathrm{d}V_g(x). \tag{2.4}$$

In words, one could say that in both sampling strategies (S1) and (S3), first x and y are sampled uniformly in M. Then in (S1), x is kept while y is resampled until $y \in D_\epsilon(x)$. In contrast, in (S3), both x and y are rejected and resampled if $(x, y) \notin \mathcal{D}_\epsilon$. These strategies are in general not the same (for example if M has a boundary). All these functionals can be observed as an instance of

$$\mathcal{E}^\epsilon[\phi] := \int_M \int_{D_\epsilon^M(x)} \left(\gamma(|\partial_{(x,y)}\phi|) + \lambda\,|\partial^2_{(x,y)}\phi|^2\right) \rho^\epsilon_{M\times M}(x, y)\,\mathrm{d}V_g(y)\,\mathrm{d}V_g(x), \tag{2.5}$$

where the density $\rho^\epsilon_{M\times M} \in L^1(M \times M)$ satisfies the following conditions:

(D1) $\int_M \int_{D_\epsilon^M(x)} \rho^\epsilon_{M\times M}(x, y)\,\mathrm{d}V_g(y)\,\mathrm{d}V_g(x) = 1$ for all $\epsilon > 0$.

(D2) There exist $c_\rho, C_\rho > 0$ with $C_\rho \geq \epsilon^m \rho^\epsilon_{M\times M}(x, y) \geq c_\rho$ for almost every $x \in M$, $y \in D_\epsilon^M(x)$, and $\epsilon > 0$.

(D3) $\epsilon^m \rho^\epsilon_{M\times M}(x, \Pi_x(\epsilon w))$ converges pointwise to some function $\rho(x, w)$ as $\epsilon \to 0$ (using the regularity of M) for almost every $x \in M, w \in B_1^m(0)$.

In fact, for all three functionals (2.2), (2.3), (2.4) the limit in (D3) is the same, the constant function $\rho(x, w) = \frac{1}{V_g(M)\omega_m}$ with $\omega_m = \mathcal{L}(B_1^m(0))$.

Remark 2.1. One may also consider more general sampling strategies, an example being sampling on an annulus $D_\epsilon^M(x) \setminus D_{c\epsilon}^M(x)$ for some constant $0 < c < 1$. In this case, the limit density will be

$$\rho(x, w) = \frac{1}{V_g(M)\omega_m(1 - c^m)} \chi_{B_1^m(0)\setminus B_c^m(0)}(w).$$

Note that the lower bound in (D2) is not satisfied in this case. However, all our results still hold, with some minor modifications in the proofs, if the lower bound assumption only holds for y from an open subset of $D_\epsilon^M(x)$ of fixed measure, for all $x \in M$.

Functionals of the type (2.5) were used in [11] in the context of (fractional) Sobolev spaces with applications in the study of variational problems, *e.g.*, in [4, 32].

Now, let us rewrite functional (2.5) using normal coordinates around each x, which will later simplify the limit analysis of the functional. To this end, we note that, for any measurable function $f_x : M \to \mathbb{R}$, we have

$$\int_{D_\epsilon^M(x)} f_x(y)\,\mathrm{d}V_g(y) = \int_{B_1^m(0)} f_x(\Pi_x(\epsilon w))w^\epsilon(x,w)\sqrt{\det G_x(\Pi_x(\epsilon w))}\epsilon^m\,\mathrm{d}w.$$

In the above, $w^\epsilon(x,w) \coloneqq \chi_{U_x^\epsilon}(\epsilon w)$, for χ_B the characteristic function of a set B, and $G_x \in \mathbb{R}^{m\times m}$ denotes the matrix representation of the Riemannian metric g, in Riemannian normal coordinates around x. Taking $f_x(y) = (\gamma(|\partial_{(x,y)}\phi|) + \lambda|\partial^2_{(x,y)}\phi|^2)\rho^\epsilon_{M\times M}(x,y)$, one obtains

$$\mathcal{E}^\epsilon[\phi] = \int_M \int_{B_1^m(0)} \left(\gamma(|\partial_{(x,\Pi_x(\epsilon w))}\phi|) + \lambda|\partial^2_{(x,\Pi_x(\epsilon w))}\phi|^2\right)\rho^\epsilon(x,w)\,\mathrm{d}w\,\mathrm{d}V_g(x) \tag{2.6}$$

for $\rho^\epsilon(x,w) \coloneqq w^\epsilon(x,w)\sqrt{\det G_x(\Pi_x(\epsilon w))}\epsilon^m\rho^\epsilon_{M\times M}(x,\Pi_x(\epsilon w))$. From the compactness of $\overline{M}$ and (D2) we deduce that there exist $\tilde{c}, \tilde{C} > 0$ with

$$\tilde{C}_\rho \geq \rho^\epsilon(x,w) \geq \tilde{c}_\rho, \qquad \text{for all } x \in M, w \in U_x, 0 < \epsilon < r_0. \tag{2.7}$$

Furthermore, using Taylor expansion of $G_x(\Pi_x(\epsilon w))$ around $\epsilon = 0$ and the local Lipschitz property of the determinant function, we have Proposition 3.1, Lemma 3.5 of [45]

$$\det G_x(\Pi_x(\epsilon w)) = 1 - \frac{1}{3}\mathrm{Ric}(\iota_x\tfrac{w}{|w|}, \iota_x\tfrac{w}{|w|})|w|^2\epsilon^2 + O(|w|^3\epsilon^3), \tag{2.8}$$

where Ric is the Ricci curvature. Due to smoothness of the metric and compactness of the manifold the remainder term is bounded uniformly in x. Together with assumption (D3), and the fact that $w^\epsilon(x,w)$ converges pointwise to 1 for every $x \in M$ and $w \in B_1^m(0)$, this implies

$$\rho^\epsilon(x,w) \to \rho(x,w) \text{ as } \epsilon \to 0, \quad \text{for a.e. } x \in M, w \in B_1^m(0). \tag{2.9}$$

3. Analysis of the continuous sampling loss functional

In this section we study the continuous sampling loss functional (2.5), or equivalently (2.6). We show existence of a minimizer in an appropriate class of functions. Furthermore, we show convergence to a local energy functional for vanishing sampling radius ϵ. (Note that the continuous sampling loss functional (2.5) is nonlocal, due to the double integral over pairs of points.)

3.1. Preliminaries on function spaces

To study the nonlocal energy (2.5), we will need suitable Riemannian notions of derivatives for functions $\phi : M \to \mathbb{R}^l$. The *Riemannian gradient (Jacobian)* $\operatorname{grad}\phi(x) = (\operatorname{grad}\phi_1(x), \dots, \operatorname{grad}\phi_l(x)) \in (T_xM)^l$ of ϕ can be defined *via* the identity

$$\frac{\mathrm{d}}{\mathrm{d}t}(\phi_j \circ \exp_x)(tv)|_{t=0} = g(\operatorname{grad}\phi_j(x), v), \quad \text{for all } v \in T_xM.$$

The Riemannian Hessian $\operatorname{Hess}\phi(x) = (\operatorname{Hess}\phi_1(x), \dots, \operatorname{Hess}\phi_l(x))$ is the linear operator $T_xM \to (T_xM)^l$ defined by $\operatorname{Hess}\phi_j(x)[v] = \nabla_v \operatorname{grad}\phi_j(x)$, $j = 1, \dots, l$, where ∇ is the Levi-Civita connection and ∇_v the covariant derivative in direction v. We have the identity [1, Chapter 5]

$$\frac{\mathrm{d}^2}{\mathrm{d}^2t}(\phi_j \circ \exp_x(tv))|_{t=0} = g_x(\operatorname{Hess}\phi_j(x)[v], v) \quad \text{for all } v \in T_xM.$$

For notational simplicity we will sometimes apply the Riemannian metric g on $(T_xM)^l \times T_xM$, which then is to be understood componentwise, *i.e.*, we use the notation $g(A,v) = (g(A_j,v))_{j=1,\dots,l}$ for an l-tuple $A=(A_1,\dots,A_l)$ of tangent vectors. Similarly, we write $g(A,B) = \sum_{j=1}^l g(A_j,B_j)$ for $A,B \in (T_xM)^l$ and $g(H,K) = \sum_{j=1}^m g(H[v_j],K[v_j])$ for linear operators $H,K : T_xM \to T_xM$, where $v_1,\dots,v_m$ is any orthonormal basis of T_xM. In the above notation the previous identity reads

$$\frac{\mathrm{d}^2}{\mathrm{d}^2 t}(\phi \circ \exp_x(tv))|_{t=0} = g(\operatorname{Hess}\phi(x)[v],v) \quad \text{for all } v \in T_xM.$$

Given $(x,y) \in \mathcal{D}_{r_0}$, denote by

$$\mathrm{PT}_{y\leftarrow x} \colon T_xM \to T_yM$$

the parallel transport operator along the unique minimizing geodesic $\gamma\colon [0,1] \to M$ with $\gamma(0)=x$ and $\gamma(1)=y$. The gradient of a differentiable function is Lipschitz of constant $L_{\mathrm{grad}}(\phi)$ if

$$\forall (x,y) \in \mathcal{D}_{r_0}, \quad \|\mathrm{PT}_{x\leftarrow y} \operatorname{grad}\phi(y) - \operatorname{grad}\phi(x)\| \le L_{\mathrm{grad}}(\phi) d_M(x,y),$$

The Hessian of a twice differentiable function ϕ is Lipschitz of constant $L_{\mathrm{Hess}}(\phi)$ if

$$\forall (x,y) \in \mathcal{D}_{r_0}, \quad \|\mathrm{PT}_{x\leftarrow y} \operatorname{Hess}\phi(y)\, \mathrm{PT}_{y\leftarrow x} - \operatorname{Hess}\phi(x)\| \le L_{\mathrm{Hess}}(\phi) d_M(x,y).$$

In both cases, $\|A\| = \sqrt{g(A,A)}$, with the summation convention as defined above. We write $\phi \in C^{2,1}(M,\mathbb{R}^l)$ for such functions (cf. [10, Section 10.4]). We have the following estimate, where from now on we use notation

$$\bar{w} \coloneqq \tfrac{w}{|w|}, \qquad \text{for any } w \in \mathbb{R}^m.$$

Lemma 3.1. *Let $\phi \in C^{2,1}(M,\mathbb{R}^l)$. Then, for all $x \in M, w \in U_x$ we have*

$$\left|\partial_{(x,\Pi_x(w))}\phi - g_x(\operatorname{grad}\phi(x), \iota_x\bar{w})\right| \le \tfrac{1}{2} L_{\mathrm{grad}}(\phi)|w|, \tag{3.1}$$

$$\left|\partial^2_{(x,\Pi_x(w))}\phi - g_x(\operatorname{Hess}\phi(x)[\iota_x\bar{w}], \iota_x\bar{w})\right| \le \tfrac{5}{6} L_{\mathrm{Hess}}(\phi)|w|. \tag{3.2}$$

Proof. Given $v \in \iota_x(U_x)$ with $\|v\|=1$, the first and second order Taylor expansion of $r \mapsto \phi\circ\exp_x(rv)$ around zero yields

$$\begin{aligned}\phi(\exp_x(rv)) &= \phi(x) + g_x(\operatorname{grad}\phi(x), rv) + R_1(x,rv)\\ &= \phi(x) + g_x(\operatorname{grad}\phi(x), rv) + \tfrac{1}{2} g_x(\operatorname{Hess}\phi(x)[rv], rv) + R_2(x,rv),\end{aligned}$$

where, following Section 10.4 of [10], the remainder terms can be bounded by $|R_1(x,rv)| \le \frac{1}{2} L_{\mathrm{grad}}(\phi) r^2$ and $|R_2(x,rv)| \le \frac{1}{6} L_{\mathrm{Hess}}(\phi) r^3$, respectively. Choosing $v = \iota_x\bar{w}$ and $r = |w|$ and $r = \frac{|w|}{2}$, respectively, we get identities

$$\begin{aligned}\partial_{(x,\Pi_x(w))}\phi - g_x(\operatorname{grad}\phi(x), \iota_x\bar{w}) &= \tfrac{R_1(x,\iota_x\bar{w})}{|w|},\\ \partial^2_{(x,\Pi_x(w))}\phi - g_x(\operatorname{Hess}\phi(x)[\iota_x\bar{w}], \iota_x\bar{w}) &= \tfrac{4R_2(x,\iota_x w) - 8R_2(x,\iota_x \frac{w}{2})}{|w|^2}.\end{aligned}$$

From these, the claim directly follows *via* the already verified bounds for the remainder terms R_1, R_2 on the right hand sides. □

The Sobolev spaces $H^1(M,\mathbb{R}^l)$ and $H^2(M,\mathbb{R}^l)$ are defined as the closure of $C^{2,1}(M,\mathbb{R}^l)$ under the norms

$$\|\phi\|^2_{H^1(M,\mathbb{R}^l)} \coloneqq \sum_{j=1}^l \int_M |\phi_j|^2 + g(\operatorname{grad}\phi_j, \operatorname{grad}\phi_j)\,\mathrm{d}V_g,$$

$$\|\phi\|^2_{H^2(M,\mathbb{R}^l)} \coloneqq \sum_{j=1}^l \int_M |\phi_j|^2 + g(\operatorname{grad}\phi_j, \operatorname{grad}\phi_j) + g(\operatorname{Hess}\phi_j, \operatorname{Hess}\phi_j)\,\mathrm{d}V_g.$$

A classical property of Sobolev spaces on a compact manifold is Rellich's embedding theorem, which implies that $H^2(M,\mathbb{R}^l)$ is compactly embedded in $H^1(M,\mathbb{R}^l)$ and in $L^2(M,\mathbb{R}^l)$. For further details on Sobolev spaces on manifolds we refer to [21] and to [5] for the Rellich embedding theorem in particular. The subset of functions with zero mean is denoted by $\dot{H}^2(M,\mathbb{R}^l)$. We will also make use of the following equivalent norms.

Proposition 3.2. *Let $V=\mathcal{C}_{1,\kappa}$, where $\mathcal{C}_{1,\kappa}\subset\mathbb{R}^m$ is a cone as defined in (M2). Then, the quantity*

$$|W|_{av} \coloneqq \left(\int_V g(W,\iota_x\bar{w})^2\,\mathrm{d}w\right)^{\frac{1}{2}} \tag{3.3}$$

for $W\in T_xM$ defines a norm on T_xM, which is uniformly equivalent to the standard norm based on the metric. Let $L_{sym}(T_xM,T_xM)$ denote the space of symmetric endomorphisms on T_xM. Then a norm on this space is given by

$$|A|_{av} \coloneqq \left(\int_V g(A[\iota_x\bar{w}],\iota_x\bar{w})^2\,\mathrm{d}w\right)^{\frac{1}{2}}, \tag{3.4}$$

and it is uniformly equivalent to the standard norm on $L_{sym}(T_xM,T_xM)$. As a consequence, the norm

$$\|\phi\|_{H^2_{av}(M,\mathbb{R}^l)} \coloneqq \left(\int_M |\operatorname{grad}\phi(x)|_{av}{}^2 + |\operatorname{Hess}\phi(x)|_{av}{}^2\,\mathrm{d}V_g(x)\right)^{\frac{1}{2}} \tag{3.5}$$

is equivalent to the standard H^2-norm on $\dot{H}^2(M,\mathbb{R}^l)$.

Proof. It is obvious that $W\mapsto|W|_{\mathrm{av}}$ is 1-homogeneous and satisfies the triangle inequality; its positive definiteness will follow from the norm equivalence shown below. Using the notation $\|W\|^2=g(W,W)$, we get $|W|_{\mathrm{av}}{}^2\le\mathcal{L}(V)\|W\|^2$ by the Cauchy–Schwarz inequality. Furthermore, using the isometry of ι_x, we observe

$$|W|_{\mathrm{av}}{}^2=\int_V(\iota_x^{-1}(W)\cdot\bar{w})^2\,\mathrm{d}w\ge c|\iota_x^{-1}(W)|^2=c\|W\|^2,$$

where $c>0$ is a constant independent of x. Namely, the second term above is a norm on $\mathbb{R}^m$. This is due to the fact that $v\cdot w=0$ for all $w\in V$ implies $v=0$, since V contains an open set. One follows an analogous argument for the norm on $L_{\mathrm{sym}}(T_xM,T_xM)$. Note that for any linear operator $B\colon\mathbb{R}^m\to\mathbb{R}^m$, $Bw\cdot w=0$ for every $w\in V$ implies that B is skew symmetric. Applying this observation to $B=\iota_x\circ A\circ\iota_x\in L_{\mathrm{sym}}(\mathbb{R}^m,\mathbb{R}^m)$, we obtain $\iota_x\circ A\circ\iota_x=0$. From this it immediately follows that the norm $\|\cdot\|^2_{H^2_{\mathrm{av}}(M,\mathbb{R}^l)}$ is equivalent to the standard $H^2(M,\mathbb{R}^l)$-seminorm $\sum_{j=1}^l\int_M g(\operatorname{grad}\phi_j,\operatorname{grad}\phi_j)+g(\operatorname{Hess}\phi_j,\operatorname{Hess}\phi_j)\,\mathrm{d}V_g$, which is equivalent to the $H^2(M,\mathbb{R}^l)$-norm defined above by Poincaré's inequality on $\dot{H}^2(M,\mathbb{R}^l)$. □

3.2. Existence of minimizer to nonlocal energy

In this section we study the existence of a minimizer for the nonlocal energy (2.5). One way to approach this problem would be to study it on the function space naturally associated with the energy $\mathcal{E}^\epsilon$. This space would be the completion of $C^{2,1}(M,\mathbb{R}^l)$ under the Hilbert space norm $\|\phi\|_\epsilon^2\coloneqq\|\phi\|^2_{L^2(M,\mathbb{R}^l)}+|\phi|_\epsilon^2$ with

$$|\phi|_\epsilon^2\coloneqq\int_M\int_{D_\epsilon^M(x)}(|\partial_{(x,y)}\phi|^2+|\partial^2_{(x,y)}\phi|^2)\rho^\epsilon_{M\times M}(x,y)\,\mathrm{d}V_g(y)\,\mathrm{d}V_g(x).$$

On this space the energy would be coercive and likely admit a minimizer; indeed (at least for our choice of γ), there exists $C>0$ such that $-C+\frac{1}{C}|\phi|_\epsilon^2\le\mathcal{E}^\epsilon[\phi]\le C+C|\phi|_\epsilon^2$, independent of ϵ. This function space is closely related to $H^2(M,\mathbb{R}^l)$: in [9, 11] it is shown that, if $M=\mathbb{R}^m$ and $\phi\in L^2(M,\mathbb{R}^l)$ satisfies $\liminf_{\epsilon\to0}|\phi|_\epsilon<\infty$, then $\phi\in H^2(M,\mathbb{R}^l)$. A corresponding result just for the first order term was also shown for smooth, compact and connected Riemannian manifolds in [29]. There are also equicoercivity or compactness results for sequences

ϕ^ϵ with uniformly bounded $\{|\phi^\epsilon|_\epsilon\}_{\epsilon>0}$ (see *e.g.* [11], Cor. 6). All this would ultimately allow us to prove that minimizers of $\mathcal{E}^\epsilon$ converge to a minimizer of a limit functional.

We will not follow this approach, and instead only look for minimizers on a more restricted ϵ-dependent set $\mathcal{F}^\epsilon$ of functions from M to $\mathbb{R}^l$, which will be realized later *via* neural networks. We prefer this latter approach for two reasons: First, we would need to translate the theory of [9,11] to functions on manifolds. Second, the compactness results are sensitive to the particular choice of the sampling weight $\rho^\epsilon_{M\times M}$. In particular, while equicoercivity will probably hold for our examples (2.2)–(2.4), one might also think of sampling weights $\rho^\epsilon_{M\times M}(x,\cdot)$ that are supported on an annulus as noted in Remark 2.1. in which case equicoercivity might potentially be lost (*cf.* [11, Counterexample 1]), while it still holds in the set $\mathcal{F}^\epsilon$.

Since the energy $\mathcal{E}^\epsilon$ is translation invariant, functions in $\mathcal{F}^\epsilon$ are (without loss of generality) assumed to have zero mean. In addition, we impose the following conditions on $\mathcal{F}^\epsilon$:

(H1) (closedness in $\dot{L}^2$) For every $0<\epsilon<r_0$, we have that $\mathcal{F}^\epsilon$ is closed as a subset of $\dot{L}^2(M,\mathbb{R}^l)$, the set of L^2 functions on M with zero mean.

(H2) (boundedness) $\mathcal{F}^\epsilon \subset C^{2,1}(M,\mathbb{R}^l)$, and there exists a constant $C_L \geq 0$ such that

$$\limsup_{\epsilon\to 0}\sup_{\phi^\epsilon\in\mathcal{F}^\epsilon} \epsilon\left(L_{\mathrm{grad}}(\phi^\epsilon)+L_{\mathrm{Hess}}(\phi^\epsilon)\right)\leq C_L,$$

(H3) (density in $\dot{H}^2(M,\mathbb{R}^l)$) For every $\phi\in\dot{H}^2(M,\mathbb{R}^l)$, there exists a sequence $\{\phi^\epsilon\}_{\epsilon>0}$ with $\phi^\epsilon\in\mathcal{F}^\epsilon$ such that $\lim_{\epsilon\to 0}\|\phi^\epsilon-\phi\|_{H^2(M,\mathbb{R}^l)}=0$.

Here the notation $\epsilon\to 0$ is short for a sequence $(\epsilon_k)_{k\in\mathbb{N}}$ with $\epsilon_k\to 0$ for $k\to\infty$. These conditions can in particular be satisfied if the functions in $\mathcal{F}^\epsilon$ are realizations of deep neural networks with sufficiently smooth nonlinear activation functions (such as the *sigmoid/logistic* function or the *softplus* function) and ϵ-dependent bounds on the network parameters and weights. Indeed, bounds on the growth of derivatives of these realizations can be computed explicitly in terms of the numbers of layers and *a priori* bounds on the network weights by applying the chain rule (*cf.* Sect. 4.2). The closedness of the spaces $\mathcal{F}^\epsilon$ was shown in Proposition 3.5 of [40], while the density property was shown in an already classical article by Hornik *et al.* [23]. Note that all these results were actually shown for neural networks defined on Euclidean domains, but they can be transferred to smooth m-dimensional manifolds (M,g) embedded in $\mathbb{R}^n$ such that the metric g is equivalent to the metric induced by the embedding. For more details we refer to Section 7 of [8]. In Section 4, we will provide more information on the specific network architecture employed by us to satisfy (H1)–(H3).

The following theorem provides the existence of a minimizer for the $\mathcal{F}^\epsilon$-restricted energy

$$\mathcal{E}^\epsilon_{\mathcal{F}}[\phi]:=\begin{cases}\mathcal{E}^\epsilon[\phi] & \text{if } \phi\in\mathcal{F}^\epsilon,\\ \infty & \text{else,}\end{cases} \tag{3.6}$$

and in addition the uniform boundedness of all minimizers in $H^2(M,\mathbb{R}^l)$, independent of ϵ.

Theorem 3.3. *Let conditions (H1)–(H2) be satisfied. Then, for every small enough $\epsilon>0$, there exists a minimizer ϕ^ϵ of energy (3.6), and $\|\phi^\epsilon\|_{H^2(M,\mathbb{R}^l)}\leq C$ for a constant $C>0$ independent of ϵ.*

Proof. Below, we will exploit the energy reformulation (2.6) of $\mathcal{E}^\epsilon$. By condition (M2) on M we have $\Pi_x(\mathcal{C}_{r_0,\kappa})\subset D^M_{r_0}(x)$, and thus in particular $\Pi_x(\mathcal{C}_{\epsilon,\kappa})\subset D^M_\epsilon(x)$ for all $\epsilon<r_0$. Let $V=\mathcal{C}_{1,\kappa}$ as in Proposition 3.2. Recalling the definition of $w^\epsilon(x,w)=\chi_{U^\epsilon_x}(\epsilon w)$, we see

$$w^\epsilon(x,w)=1 \text{ for all } x\in M, w\in V. \tag{3.7}$$

Using first (2.7) (which comes from (D2)) and then the inequality $|a|^2 \geq \frac{1}{2}|b|^2 - |a-b|^2$ for $a = \partial_{(x,\Pi_x(\epsilon w))}\phi$ and $b = g_x(\operatorname{grad}\phi(x), \iota_x w)$, we get

$$\begin{aligned}
&\int_M \int_{D_\epsilon^M(x)} |\partial_{(x,y)}\phi|^2 \rho^\epsilon_{M\times M}(x,y)\,\mathrm{d}V_g(y)\,\mathrm{d}V_g(x)\\
&= \int_M \int_{B_1^m(0)} |\partial_{(x,\Pi_x(\epsilon w))}\phi|^2 \rho^\epsilon(x,w)\,\mathrm{d}w\,\mathrm{d}V_g(x)\\
&\geq \frac{\tilde{c}_\rho}{2}\int_M\int_V |g_x(\operatorname{grad}\phi(x), \iota_x\bar{w})|^2\,\mathrm{d}w\,\mathrm{d}V_g(x)\\
&\quad - \tilde{c}_\rho \int_M\int_V |g_x(\operatorname{grad}\phi(x), \iota_x\bar{w}) - \partial_{(x,\Pi_x(\epsilon w))}\phi|^2\,\mathrm{d}w\,\mathrm{d}V_g(x)\\
&\geq \frac{\tilde{c}_\rho}{2}\int_M |\operatorname{grad}\phi|_{\mathrm{av}}{}^2\,\mathrm{d}V_g - C\epsilon^2 L_{\operatorname{grad}}(\phi)^2
\end{aligned}$$

for a constant $C > 0$, where in the last step we used Lemma 3.1. We analogously obtain

$$\int_M \int_{D_\epsilon^M(x)} |\partial^2_{(x,y)}\phi|^2 \rho^\epsilon_{M\times M}(x,y)\,\mathrm{d}V_g(y)\,\mathrm{d}V_g(x) \geq \frac{\tilde{c}_\rho}{2}\int_M |\operatorname{Hess}\phi|_{\mathrm{av}}{}^2\,\mathrm{d}V_g - C\epsilon^2 L_{\operatorname{Hess}}(\phi)^2.$$

Summing both terms and applying Proposition 3.2, one sees that there exists $C > 0$ such that

$$\|\phi\|^2_{H^2(M,\mathbb{R}^l)} \leq C\left(\mathcal{E}^\epsilon[\phi] + C + \epsilon^2\left(L_{\operatorname{grad}}(\phi) + L_{\operatorname{Hess}}(\phi)\right)^2\right) \leq C\left(\mathcal{E}^\epsilon[\phi] + C + C_L^2\right), \tag{3.8}$$

for every $\phi \in \mathcal{F}^\epsilon$ and ϵ small enough (*cf.* assumption (H2)). Obviously, $\mathcal{E}^\epsilon[0]$ is a finite upper bound for the energy of the minimizer. Thus, for any fixed $\epsilon > 0$, there exists a minimizing sequence $\{\phi_j^\epsilon\}_{j\in\mathbb{N}}$ with $\|\phi_j^\epsilon\|_{H^2(M,\mathbb{R}^l)} \leq C(\mathcal{E}^\epsilon[0] + C_L^2)$ and $\lim_{j\to\infty}\mathcal{E}^\epsilon[\phi_j^\epsilon] = \inf_{\phi\in\mathcal{F}^\epsilon}\mathcal{E}^\epsilon[\phi]$. Hence, there exists a weakly convergent subsequence (after relabeling) $\phi_j^\epsilon \rightharpoonup \phi^\epsilon$ in $H^2(M,\mathbb{R}^l)$. By weak lower semicontinuity of the norm, we have $\|\phi^\epsilon\|_{H^2(M,\mathbb{R}^l)} \leq C(\mathcal{E}^\epsilon[0] + C_L^2)$. Furthermore, we have $\phi_j^\epsilon \to \phi^\epsilon$ in $\dot{L}^2(M,\mathbb{R}^l)$ by the compact embedding property, so that (H1) implies $\phi^\epsilon \in \mathcal{F}^\epsilon$. Extracting a further subsequence, we even obtain pointwise almost everywhere convergence, so that by Fatou's lemma we have $\liminf_{j\to\infty}\mathcal{E}^\epsilon[\phi_j^\epsilon] \geq \mathcal{E}^\epsilon[\phi^\epsilon] = \mathcal{E}^\epsilon_{\mathcal{F}}[\phi^\epsilon]$, which finishes the proof. □

Uniqueness of a minimizer to (3.6) cannot be expected: Indeed, the energy $\mathcal{E}^\epsilon$ is invariant under composition of its argument from the left with a rigid motion or a reflection. Even apart from this invariance, the nonconvexity of the first integrand in (2.5), which is unavoidable when promoting isometries, may prevent uniqueness. However, whenever M is intrinsically flat and homeomorphic to the m-disc, there is a unique minimizer of $\mathcal{E}^\epsilon$ up to rigid motion and reflection Proposition 1 of [14]. Being able to find flat embeddings may actually be quite relevant in applications, as it was noticed in [47] that generative image manifolds have almost no curvature.

3.3. The limit for vanishing sampling radius

In this section we study convergence of the functional (3.6) as the sampling radius ϵ tends to 0. The limit loss functional is local, promotes low distortion and low bending embeddings, and reads

$$\mathcal{E}[\phi] \coloneqq \int_M \Gamma(\operatorname{grad}\phi(x)) + \lambda\|\operatorname{Hess}\phi(x)\|^2\,\mathrm{d}V_g(x), \qquad \text{with} \tag{3.9}$$

$$\Gamma(W) \coloneqq \int_{B_1^m(0)} \gamma(|g_x(W,\iota_x\bar{w})|)\rho(x,w)\,\mathrm{d}w\,, \tag{3.10}$$

$$\|A\|^2 \coloneqq \int_{B_1^m(0)} |g_x(A[\iota_x\bar{w}],\iota_x\bar{w})|^2\rho(x,w)\,\mathrm{d}w\,, \tag{3.11}$$

for all $W \in T_xM$ and $A \in L(T_xM, T_xM)$ and ρ from (D3), as stated in our main result.

Theorem 3.4 (Mosco-convergence)**.** *Let (H2–H3) be satisfied with $C_L = 0$. Then the nonlocal regularization energies $\{\mathcal{E}^\epsilon_{\mathcal{F}}\}_{\epsilon>0}$ given by (3.6) converge to the continuous regularization energy $\mathcal{E}$ given by (3.9) as $\epsilon \to 0$, in the sense of Mosco in $\dot{H}^2(M,\mathbb{R}^l)$.*

Remark 3.5. An isometric embedding $\phi : M \to \mathbb{R}^l$ is characterized by $\operatorname{grad}\phi(x)$ being orthogonal in any point $x \in M$, or equivalently $|g_x(\operatorname{grad}\phi(x), \iota_x v)| = 1$ for all $v \in S^{m-1}$, where $S^{m-1} \coloneqq \partial B^m_1(0)$. Therefore, the first term of (3.9) penalizes deviation from an isometric embedding, as it is zero for locally isometric $\phi : M \to \mathbb{R}^l$ and strictly positive otherwise. Similarly, extrinsic bending of the embedding $\phi : M \to \mathbb{R}^l$ manifests as a non-vanishing Riemannian Hessian. Since the term (3.11) defines a squared norm on the space $L_{\mathrm{sym}}(T_xM, T_xM)^l$ by Proposition 3.2, the second term in (3.9) penalizes extrinsic bending.

Proof of Theorem 3.4. We have to verify the two properties [35] defining Mosco-convergence:

lim inf: For every sequence $\{\phi^\epsilon\}_{\epsilon>0}$ converging weakly to ϕ in $\dot{H}^2(M,\mathbb{R}^l)$ as $\epsilon \to 0$, one has that $\liminf_{\epsilon\to 0} \mathcal{E}^\epsilon_{\mathcal{F}}[\phi^\epsilon] \geq \mathcal{E}[\phi]$.

lim sup: For every $\phi \in \dot{H}^2(M,\mathbb{R}^l)$, there exists a sequence $\{\phi^\epsilon\}_{\epsilon>0}$ converging for $\epsilon \to 0$ strongly to ϕ in $\dot{H}^2(M,\mathbb{R}^l)$ with $\limsup_{\epsilon\to 0} \mathcal{E}^\epsilon_{\mathcal{F}}[\phi^\epsilon] \leq \mathcal{E}[\phi]$.

For ease of notation, we will sometimes use an expression of the form $f(x,w)$ to actually indicate the mapping $M \times B^m_1(0) \ni (x,w) \mapsto f(x,w)$. It will always be clear from the context when this is intended. For easier reference, we also recall the form of our nonlocal functional

$$\mathcal{E}^\epsilon[\phi] = \int_M \int_{B^m_1(0)} \left(\gamma(|\partial_{(x,\Pi_x(\epsilon w))}\phi|) + \lambda|\partial^2_{(x,\Pi_x(\epsilon w))}\phi|^2\right) \rho^\epsilon(x,w)\,\mathrm{d}w\,\mathrm{d}V_g(x). \tag{3.12}$$

Proof of the lim inf *property.*

Let $\phi^\epsilon \rightharpoonup \phi$ in $\dot{H}^2(M,\mathbb{R}^l)$. Putting aside the trivial case of $\liminf_{\epsilon\to 0} \mathcal{E}^\epsilon_{\mathcal{F}}[\phi^\epsilon] = +\infty$, we may assume that $\phi^\epsilon \in \mathcal{F}^\epsilon$ for every $\epsilon > 0$ along a subsequence. We first estimate

$$\begin{aligned}
&\limsup_{\epsilon\to 0} \|\partial_{(x,\Pi_x(\epsilon w))}\phi^\epsilon - g_x(\operatorname{grad}\phi(x), \iota_x\bar{w})\|_{L^2(M\times B^m_1(0))} \\
&\leq \limsup_{\epsilon\to 0} \|\partial_{(x,\Pi_x(\epsilon w))}\phi^\epsilon - g_x(\operatorname{grad}\phi^\epsilon(x), \iota_x\bar{w})\|_{L^2(M\times B^m_1(0))} \\
&\quad + \limsup_{\epsilon\to 0} \|g_x(\operatorname{grad}\phi^\epsilon(x), \iota_x\bar{w}) - g_x(\operatorname{grad}\phi(x), \iota_x\bar{w})\|_{L^2(M\times B^m_1(0))}.
\end{aligned}$$

By (3.1) the first summand can be estimated as

$$\|\partial_{(x,\Pi_x(\epsilon w))}\phi^\epsilon - g_x(\operatorname{grad}\phi^\epsilon(x), \iota_x\bar{w})\|_{L^2(M\times B^m_1(0))} \leq C\epsilon L_{\mathrm{grad}}(\phi^\epsilon),$$

which by (H2) converges to 0. To show the same for the second summand, we observe that $\operatorname{grad}\phi^\epsilon \to \operatorname{grad}\phi$ in $L^2(M,(TM)^l)$ by the compact embedding of $\dot{H}^2(M,\mathbb{R}^l)$ into $H^1(M,\mathbb{R}^l)$. Thus we obtain $g_x(\operatorname{grad}\phi^\epsilon(x), \iota_x\bar{w}) \to g_x(\operatorname{grad}\phi(x), \iota_x\bar{w})$ in $L^2(M \times B^m_1(0), \mathbb{R}^l)$ since the function

$$V \mapsto [(x,w) \mapsto g_x(V(x), \iota_x\bar{w})]$$

is a bounded linear function from $L^2(M,(TM)^l)$ to $L^2(M \times B^m_1(0), \mathbb{R}^l)$. Hence, exploiting (2.9) and passing to a subsequence without relabeling, for almost every $x \in M, w \in B^m_1(0)$ we have

$$\gamma(|\partial_{(x,\Pi_x(\epsilon w))}\phi^\epsilon|)\rho^\epsilon(x,w) \to \gamma(|g_x(\operatorname{grad}\phi(x), \iota_x\bar{w})|)\rho(x,w).$$

Applying Fatou's lemma we obtain

$$\begin{aligned}&\liminf_{\epsilon\to 0}\int_M\int_{B_1^m(0)}\gamma(|\partial_{(x,\Pi_x(\epsilon w))}\phi^\epsilon|)\rho^\epsilon(x,w)\,\mathrm{d}w\,\mathrm{d}V_g(x)\\&\geq\int_M\int_{B_1^m(0)}\gamma(|g_x(\operatorname{grad}\phi(x),\iota_x\bar{w})|)\rho(x,w)\,\mathrm{d}w\,\mathrm{d}V_g(x).\end{aligned}\tag{3.13}$$

To handle the second order term in (3.12), we write

$$\begin{aligned}&\liminf_{\epsilon\to 0}\|\partial^2_{(x,\Pi_x(\epsilon w))}\phi^\epsilon\sqrt{\rho^\epsilon(x,w)}\|_{L^2(M\times B_1^m(0))}\\&\geq\liminf_{\epsilon\to 0}\|g_x(\operatorname{Hess}\phi^\epsilon(x)[\iota_x\bar{w}],\iota_x\bar{w})\sqrt{\rho^\epsilon(x,w)}\|_{L^2(M\times B_1^m(0))}\\&\quad-\limsup_{\epsilon\to 0}\left\|\left(\partial^2_{(x,\Pi_x(\epsilon w))}\phi^\epsilon-g_x(\operatorname{Hess}\phi^\epsilon(x)[\iota_x\bar{w}],\iota_x\bar{w})\right)\sqrt{\rho^\epsilon(x,w)}\right\|_{L^2(M\times B_1^m(0))}.\end{aligned}$$

By (3.2) and (2.7) the last term can be estimated as

$$\left\|\left(\partial^2_{(x,\Pi_x(\epsilon w))}\phi^\epsilon-g_x(\operatorname{Hess}\phi^\epsilon(x)[\iota_x\bar{w}],\iota_x\bar{w})\right)\sqrt{\rho^\epsilon(x,w)}\right\|_{L^2(M\times B_1^m(0))}\leq C\epsilon L_{\mathrm{Hess}}(\phi^\epsilon),$$

which vanishes in the limit by (H2). To bound the first term, we first observe that

$$g_x(\operatorname{Hess}\phi^\epsilon(x)[\iota_x\bar{w}],\iota_x\bar{w})\rightharpoonup g_x(\operatorname{Hess}\phi(x)[\iota_x\bar{w}],\iota_x\bar{w})\quad\text{in } L^2(M\times B_1^m(0),\mathbb{R}^l),$$

since the function

$$A\mapsto[(x,w)\mapsto g_x(A(x)[\iota_x\bar{w}],\iota_x\bar{w})]$$

is a bounded linear function from $L^2(M,L_{\mathrm{sym}}(TM,TM)^l)$ to $L^2(M\times B_1^m(0),\mathbb{R}^l)$ and $\operatorname{Hess}\phi^\epsilon$ converges weakly to $\operatorname{Hess}\phi$ in the former space.

Furthermore, by the uniform boundedness (2.7) and pointwise convergence (2.9) of ρ^ϵ , using the dominated convergence theorem, we get

$$g_x(\operatorname{Hess}\phi^\epsilon(x)[\iota_x\bar{w}],\iota_x\bar{w})\sqrt{\rho^\epsilon(x,w)}\rightharpoonup g_x(\operatorname{Hess}\phi(x)[\iota_x\bar{w}],\iota_x\bar{w})\sqrt{\rho(x,w)}\tag{3.14}$$

in $L^2(M\times B_1^m(0),\mathbb{R}^l)$. Using the weak lower semicontinuity of the norm, this implies

$$\begin{aligned}&\liminf_{\epsilon\to 0}\|g_x(\operatorname{Hess}\phi^\epsilon(x)[\iota_x\bar{w}],\iota_x\bar{w})\sqrt{\rho^\epsilon(x,w)}\|_{L^2(M\times B_1^m(0))}\\&\geq\|g_x(\operatorname{Hess}\phi(x)[\iota_x\bar{w}],\iota_x\bar{w})\sqrt{\rho(x,w)}\|_{L^2(M\times B_1^m(0))}.\end{aligned}$$

We thus proved

$$\begin{aligned}&\liminf_{\epsilon\to 0}\int_M\int_{B_1^m(0)}\lambda|\partial^2_{(x,\Pi_x(\epsilon w))}\phi^\epsilon|^2\rho^\epsilon(x,w)\,\mathrm{d}w\,\mathrm{d}V_g(x)\\&\geq\int_M\int_{B_1^m(0)}\lambda|g_x(\operatorname{Hess}\phi[\iota_x\bar{w}],\iota_x\bar{w})|^2\rho(x,w)\,\mathrm{d}w\,\mathrm{d}V_g(x).\end{aligned}$$

Finally, combining this last inequality with (3.13), we get the desired inequality.

Proof of the lim sup *property.*

By the density assumption (H3), for every $\phi \in \dot{H}^2(M, \mathbb{R}^l)$ there exists a sequence $\{\phi^\epsilon\}_{\epsilon>0}$ such that $\phi^\epsilon \in \mathcal{F}^\epsilon$ and $\phi^\epsilon \to \phi$ in $H^2(M, \mathbb{R}^l)$ as $\epsilon \to 0$. Then we can repeat the argument from the lim inf property to prove that

$$\partial_{(x,\Pi_x(\epsilon w))}\phi^\epsilon \to g_x(\operatorname{grad}\phi(x), \iota_x\bar{w}) \text{ in } L^2(M \times B_1^m(0)).$$

Now, one uses uniform boundedness (2.7) and pointwise convergence (2.9) of ρ^ϵ and, taking into account the splitting of $\gamma(s)$ into $|s|^2$ and the uniformly bounded $\gamma(s) - |s|^2$, applies the dominated convergence theorem to obtain

$$\begin{aligned}&\limsup_{\epsilon\to 0} \int_M \int_{B_1^m(0)} \gamma(|\partial_{(x,\Pi_x(\epsilon w))}\phi^\epsilon|)\rho^\epsilon(x,w)\,\mathrm{d}w\,\mathrm{d}V_g(x) \\ &= \int_M \int_{B_1^m(0)} \gamma(|g_x(\operatorname{grad}\phi(x), \iota_x\bar{w})|)\rho(x,w)\,\mathrm{d}w\,\mathrm{d}V_g(x).\end{aligned} \tag{3.15}$$

For the second order term in (3.12) we write

$$\begin{aligned}&\limsup_{\epsilon\to 0} \|\partial^2_{(x,\Pi_x(\epsilon w))}\phi^\epsilon \sqrt{\rho^\epsilon(x,w)}\|_{L^2(M\times B_1^m(0))} \\ &\le \limsup_{\epsilon\to 0} \|g_x(\operatorname{Hess}\phi^\epsilon(x)[\iota_x\bar{w}], \iota_x\bar{w})\sqrt{\rho^\epsilon(x,w)}\|_{L^2(M\times B_1^m(0))} \\ &\quad + \limsup_{\epsilon\to 0} \| \left(\partial^2_{(x,\Pi_x(\epsilon w))}\phi^\epsilon - g_x(\operatorname{Hess}\phi^\epsilon(x)[\iota_x\bar{w}], \iota_x\bar{w})\right) \sqrt{\rho^\epsilon(x,w)}\|_{L^2(M\times B_1^m(0))},\end{aligned}$$

where the last term vanishes as in the proof of the lim inf property. For the first term we show

$$g_x(\operatorname{Hess}\phi^\epsilon(x)[\iota_x\bar{w}], \iota_x\bar{w})\sqrt{\rho^\epsilon(x,w)} \to g_x(\operatorname{Hess}\phi(x)[\iota_x\bar{w}], \iota_x\bar{w})\sqrt{\rho(x,w)}$$

in $L^2(M \times B_1^m(0), \mathbb{R}^l)$, with an analogous argument to the one used to obtain (3.14). Thus, again using the dominated convergence theorem, we finally get

$$\begin{aligned}&\limsup_{\epsilon\to 0} \int_M \int_{B_1^m(0)} \lambda|\partial^2_{(x,\Pi_x(\epsilon w))}\phi^\epsilon|^2\rho^\epsilon(x,w)\,\mathrm{d}w\,\mathrm{d}V_g(x) \\ &\le \int_M \int_{B_1^m(0)} \lambda|g_x(\operatorname{Hess}\phi(x)[\iota_x\bar{w}], \iota_x\bar{w})|^2\rho(x,w)\,\mathrm{d}w\,\mathrm{d}V_g(x),\end{aligned}$$

which together with (3.15) proves the inequality and finishes the proof of the theorem. □

The existence of a minimizer for (3.9) can be established using the direct method of the calculus of variations Theorem 2 of [14]. As for $\mathcal{E}^\epsilon_\mathcal{F}$, the minimizer (modulo a rigid motion or reflection) is in general not unique due to the isometry promoting term. Based on the existence and boundedness result from Theorem 3.3 and the Mosco-convergence we can show convergence of minimizers by a standard procedure in Γ-convergence theory (see *e.g.* [12]).

Corollary 3.6. *Let (H1)–(H3) hold with $C_L = 0$. Then for every sequence $\{\phi^\epsilon\}_{\epsilon>0}$ of minimizers for the energies $\{\mathcal{E}^\epsilon_\mathcal{F}\}_{\epsilon>0}$ there exists a subsequence converging weakly in $H^2(M, \mathbb{R}^l)$ to a minimizer ϕ for the energy $\mathcal{E}$. Furthermore, the corresponding function values $\{\mathcal{E}^\epsilon_\mathcal{F}[\phi^\epsilon]\}_{\epsilon>0}$ converge to $\mathcal{E}[\phi]$.*

4. Numerical experiments

As a testbed, we purposely use image manifolds where the manifold is explicitly known and where av_M and d_M can be explicitly computed. Note that for various more general shape manifolds with application in

imaging, algorithmic tools to compute suitable approximations of av_M and d_M are available. Examples of such shape spaces are LDDMM [54], metamorphosis [50], or optimal transport [41]. For two reasons, using these tools for training the encoder is beyond the scope of this article: First, these methods are still computationally demanding, such that it would be quite involved to even generate a moderately large set of training data. To overcome this difficulty one could replace the local distance computation and the local Riemannian average by computationally more efficient approximations. This would require a careful analysis of the approximation error. Second, while the input data may lie on a low dimensional hidden manifold, the methods will in general not compute intrinsic distances and averages within this manifold. Hence, our analysis would have to be adapted to such perturbed, non exact training data.

4.1. Different input manifolds

In the same spirit as [16], we consider image data that implicitly represent four different manifolds:

Sundial Shadows (**S**). This dataset has the upper hemisphere $S^2 \cap \{x_3 \geq 0\}$ as its underlying hidden manifold, corresponding to possible positions of a lightsource. Inspired by [37], we generate images by casting a shadow of a vertical rod on a plane from all these positions. These images form our input manifold M with metric induced from S^2, so that the distance on M is the geodesic distance on S^2, $d_M(x,y) = \arccos(x^T y)$ for $x, y \in S^2$. Contrary to [37], we do not render these images with a 3D engine, but simply approximate the shadows by Gaussians (*cf.* Fig. 1): A point $x \in S^2$ is first mapped onto the plane by mapping it onto the point $x_p \in \mathbb{R}^2$ defined as the intersection of the plane with the line through x and the rod tip. We then use a Gaussian function centered at $x_p/2$ with variance $|x_p|^2$ in direction x_p and a fixed small variance in the orthogonal direction. We use a resolution of 64×64 pixels and sampling strategy (S3).

Rotating Object (**R**). This dataset has the group of rotations $SO(3)$ as hidden manifold. We generate images forming the manifold M by projecting an arbitrarily rotated three-dimensional object (here a toy cow model[1]) to the viewing plane. We use `Pytorch3D` [43] to render the images for the training. We use the induced distance from $SO(3)$ equipped with the metric $g_x(v,w) = \frac{1}{2}\operatorname{tr}(v^T w)$ for $x \in SO(3)$, $v, w \in T_x SO(3)$ as distance on M. This geodesic distance can be computed for example *via* quaternions as $d_M(x,y) = 2\arccos(|q(x) \cdot q(y)|)$ [24], for $x, y \in SO(3)$ and $q(x), q(y)$ their representation as quaternions. We consider an image resolution of 128×128 pixels and sampling strategy (S2), which can also be implemented *via* the quaternion representation.

Arc rotating around two specified axes (**A**). This dataset has the Klein bottle $[0,1]^2/\sim$ as the hidden manifold, where $\sim$ represents the glueing operation on the boundary of $[0,1]^2$ defining the Klein bottle, *i.e.* we identify $(x_1, 0)$ and $(x_1, 1)$ as well as $(0, x_2)$ and $(1, 1 - x_2)$. To generate the image representation of $(x_1, x_2) \in [0,1]^2/\sim$ as a point in the manifold M, we use Blender[2] to render a bent, capped tube representing the arc $[0,1] \ni t \mapsto (\cos(t\pi), \sin(t\pi), 0)$, rotated by the two angles $\alpha = \pi x_1$ and $\beta = \pi(2x_2 + \frac{1}{2})$ (*cf.* Fig. 7, top row). The tube is colored using a transition from purple inside to yellow outside. We use the flat Euclidean geometry, sampling strategy (S3) on $[0,1]^2/\sim$ and an image resolution of 128×128 pixels.

Ellipses (**E**). We consider rotations and translations of an ellipse of fixed aspect ratio $a > 0$ and scale $s > 0$ given by the implicit equation $h_a(z/s) \leq 0$ with $h_a(z) = z_1^2 + (az_2)^2 - 1$ for $z \in \mathbb{R}^2$. We define a rotation about $\theta \in [0, 2\pi)$ by $R(\theta) = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix}$. For $\theta \in [0, 2\pi)$ and a translation $z_c \in [-1,1]^2$, we then consider the ellipse

$$E(\theta, z_c) = \left\{ z \in \mathbb{R}^2 \,:\, f_{\theta,z_c}(z) \coloneqq h_a\left(\tfrac{1}{s} R\left(\tfrac{\theta}{2}\right)(z - z_c)\right) \leq 0 \right\}. \tag{4.1}$$

The underlying hidden manifold of these ellipses is $[0, 2\pi)/\sim \times [-1,1]^2 \cong S^1 \times [-1,1]^2$, where $\sim$ denotes identification of 0 and 2π. The corresponding squared distance between $x = (\theta_x, z_{c,x})$ and $y = (\theta_y, z_{c,y})$ is

$$d_M(x,y)^2 = \min(|\theta_x - \theta_y|, |\theta_x + 2\pi - \theta_y|, |\theta_x - 2\pi - \theta_y|)^2 + |z_{c,x} - z_{c,y}|^2.$$

[1] https://www.cs.cmu.edu/~kmcrane/Projects/ModelRepository/

[2] https://www.blender.org/

The ellipses are discretized as 64×64 images by evaluating either the characteristic function of $E(\theta, x_c)$ or the smoothed variant

$$\hat{f}_{\theta,x_c;k}(z) = (1 + \exp(k f_{\theta,x_c}))^{-1}$$

on a grid on $[-1,1]^2$. We use a modified version of sampling strategy (S1), where we sample on the disc with radius $\frac{1}{\sqrt{3}}\epsilon$ taken with respect to the maximum metric.

4.2. Autoencoder architecture and training procedure

The used architecture is a deep convolutional neural network as in [7], however, we use larger input images and a smaller latent space dimension. Both encoder and decoder consist of blocks of two convolutions, each followed by a nonlinearity, and a subsequent average pooling (encoder) or upsampling (decoder). We determine the number of blocks such that the final output image of the encoder consists of 4×4 pixels, resulting in a latent code of size 16. The full architectures are shown in Table 1. We use Kaiming initialization [20], *i.e.*, all convolutional weights are initialized as zero-mean Gaussian random variables with standard deviation $\sqrt{2}/\sqrt{\texttt{fan\_in}(1+0.01^2)}$ for `fan_in` the number of input channels of the layer times the number of entries in the filter, and all biases are initialized with zeros. For training, we use the Adam optimizer [27] with learning rate 10^{-4} and default values $\beta_1 = 0.9, \beta_2 = 0.999$ and $\varepsilon = 10^{-8}$. We further employ a so-called weight decay (an L^2-penalty on the weights) of 10^{-5}. In [14], we use the non-differentiable leaky ReLU activation function $\mathrm{LeakyReLU}_\alpha(x) = \max\{x, \alpha x\}$. To satisfy the assumption (H2), one may replace this activation function with a smooth approximation $\mathrm{LeakyReLU}_{\alpha,\beta}(x) = \frac{1}{\beta}\log(\exp(\beta x) + \exp(\alpha\beta x))$, which converges to $\mathrm{LeakyReLU}_\alpha$ for $\beta \to \infty$. Since the leaky ReLU is much faster to compute and we did not observe any significant effect of the smoothing in practice, we keep the leaky ReLU activation function. Networks with (leaky) ReLU nonlinearity also provide nice approximation properties with provable growth rate of the network architecture, though only in Sobolev spaces of order $0 \le s \le 1$ [18]. Furthermore, in view of (H3), the architecture (number of layers and weights, bound on weight growth or penalty on weights) should depend on the sampling radius ϵ. However, since our experiments are performed with ϵ in a fixed range, we always keep the same architecture.

The training data consists of triplets of images together with a distance value (*cf.* Fig. 1 top):

$$(x, y, \mathrm{av}_M(x,y), d_M(x,y)) \text{ with } x, y \in \mathcal{S}_\epsilon \subset M \times M.$$

This input allows to compute the ingredients $\partial_{(x,y)}\phi$ and $\partial^2_{(x,y)}\phi$ of the discrete sampling loss functional $E^{\mathcal{S}_\epsilon}[\phi]$ defined in Section 2. Even though we generate random samples on the fly, we simulate epochs of size 10000, split into batches of size 128, to make the training graphs look more familiar and less noisy.

Our method allows us to train the encoder map separately from the decoder by minimizing $E^{\mathcal{S}_\epsilon}[\phi_\theta]$ to obtain some close to optimal ϕ_{θ^*}. Subsequently, if required, one can fix θ^* and train the decoder by minimizing the *reconstruction loss*

$$R[\psi_\xi] = R[\phi_{\theta^*}, \psi_\xi] = \frac{1}{2|\mathcal{S}_\epsilon|} \sum_{(x,y)\in\mathcal{S}_\epsilon} \|\psi_\xi(\phi_{\theta^*}(x)) - x\|^2_{L^2} + \|\psi_\xi(\phi_{\theta^*}(y)) - y\|^2_{L^2}, \tag{4.2}$$

where for a pixel image $u \in \mathbb{R}^{N\times N\times C}$ with $N \times N$ pixels and C color channels we define the discrete L^2-norm as $\|u\|^2_{L^2} := \frac{1}{N^2 C}\sum_{i,j=1}^{N}\sum_{c=1}^{C} u^2_{ijc}$. Alternatively, encoder and decoder can be trained simultaneously by minimizing the loss functional

$$E^{\mathcal{S}_\epsilon}[\phi_\theta] + \kappa R[\phi_\theta, \psi_\xi]$$

for ϕ_θ and ψ_ξ and some $\kappa > 0$. From now on, we omit the parameters θ and ξ again, and write ϕ, ψ for encoder and decoder. For better numerical stability in the computation of the finite differences $\partial_{(x,y)}\phi$ and $\partial^2_{(x,y)}\phi$, we rejected pairs with distance below a certain threshold. In all our experiments, the isometry, flatness and reconstruction parts of the loss function decrease continuously and monotonically (up to the usual stochastic variations) during training, as shown in Figure 3 for the dataset (**R**).

TABLE 1. Encoder (left) and decoder (right) architectures for 64×64 grayscale images. Parameters for convolutional layers (Conv2d) are filter size/stride/padding/output channels, where filter size, stride and padding are in each direction. Parameters for average pooling layers (AvgPool2d) are filter size and stride, again in each direction. Parameters for upsampling layers (Upsample) are scale and mode. For the LeakyReLU activation, a negative slope of $\alpha = 0.01$ was used. Output sizes are given with channels in the last dimension. The architecture for 128×128 3-channel color images is similar, except that there is one more block, thus the maximum number of output channels of the encoder is 512.

Type	Parameters	Output size
Conv2d	1/1/1/16	$66 \times 66 \times 16$
Conv2d + LeakyReLU	3/1/1/16	$66 \times 66 \times 16$
Conv2d + LeakyReLU	3/1/1/16	$66 \times 66 \times 16$
AvgPool2d	2/2	$33 \times 33 \times 16$
Conv2d + LeakyReLU	3/1/1/32	$33 \times 33 \times 32$
Conv2d + LeakyReLU	3/1/1/32	$33 \times 33 \times 32$
AvgPool2d	2/2	$16 \times 16 \times 32$
Conv2d + LeakyReLU	3/1/1/64	$16 \times 16 \times 64$
Conv2d + LeakyReLU	3/1/1/64	$16 \times 16 \times 64$
AvgPool2d	2/2	$8 \times 8 \times 64$
Conv2d + LeakyReLU	3/1/1/128	$8 \times 8 \times 128$
Conv2d + LeakyReLU	3/1/1/128	$8 \times 8 \times 128$
AvgPool2d	2/2	$4 \times 4 \times 128$
Conv2d + LeakyReLU	3/1/1/256	$4 \times 4 \times 256$
Conv2d	3/1/1/1	$4 \times 4 \times 1$

Type	Parameters	Output size
Conv2d + LeakyReLU	3/1/1/128	$4 \times 4 \times 128$
Conv2d + LeakyReLU	3/1/1/128	$4 \times 4 \times 128$
Upsample	2 / nearest	$8 \times 8 \times 128$
Conv2d + LeakyReLU	3/1/1/64	$8 \times 8 \times 64$
Conv2d + LeakyReLU	3/1/1/64	$8 \times 8 \times 64$
Upsample	2 / nearest	$16 \times 16 \times 64$
Conv2d + LeakyReLU	3/1/1/32	$16 \times 16 \times 32$
Conv2d + LeakyReLU	3/1/1/32	$16 \times 16 \times 32$
Upsample	2 / nearest	$32 \times 32 \times 32$
Conv2d + LeakyReLU	3/1/1/16	$32 \times 32 \times 16$
Conv2d + LeakyReLU	3/1/1/16	$32 \times 32 \times 16$
Upsample	2 / nearest	$64 \times 64 \times 16$
Conv2d + LeakyReLU	3/1/1/16	$64 \times 64 \times 16$
Conv2d	3/1/1/1	$64 \times 64 \times 1$

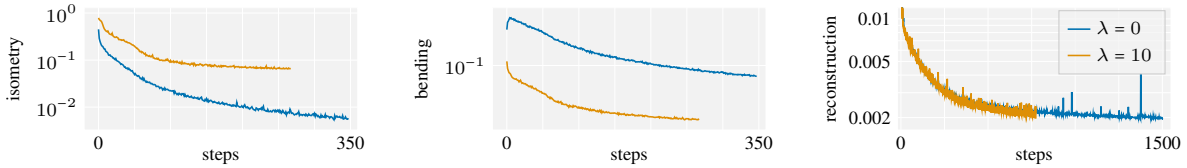

FIGURE 3. Comparison of temporal evolution of the three loss components for dataset (**R**) for $\lambda = 10$ and $\lambda = 0$ (logarithmic y-axis). Per optimization step, 10000 images are processed in batches of 128.

4.3. Visualization of the embeddings

To evaluate the smoothness of our embedding function, we visualize the embedded manifold $\phi(M)$ by performing a principal component analysis (PCA) in latent space. We then visualize projections onto three-dimensional subspaces spanned by three selected principal directions of the PCA. In all cases we observe smooth embeddings that neatly reproduce the geometry and topology of the hidden manifold M (see Figs. 1, 4, 7, 5 and 10). Note that self-intersections in the three-dimensional projection do not necessarily reflect self-intersections in the full latent spaces (see Figs. 6 and 7). For dataset (**S**), our approach and its result are summarized in Figure 1 for $\lambda = 0$. The geometry of the hemisphere is reproduced. In comparison, Figure 4 shows the result for a higher bending penalty $\lambda = 10$, resulting in a much flatter embedding of the hemisphere (*i.e.*, the variance in the third component is significantly smaller).

In all our experiments, the latent space dimensionality l is chosen substantially larger than the minimum value required for a smooth embedding – in real applications the intrinsic dimensionality m is unknown beforehand anyway, so that one needs to pick l rather larger than smaller. Furthermore, the encoder might achieve flatter embeddings when more dimensions are available. For instance, $S^1 \times S^1 \subset \mathbb{R}^4$ provides a better embedding of

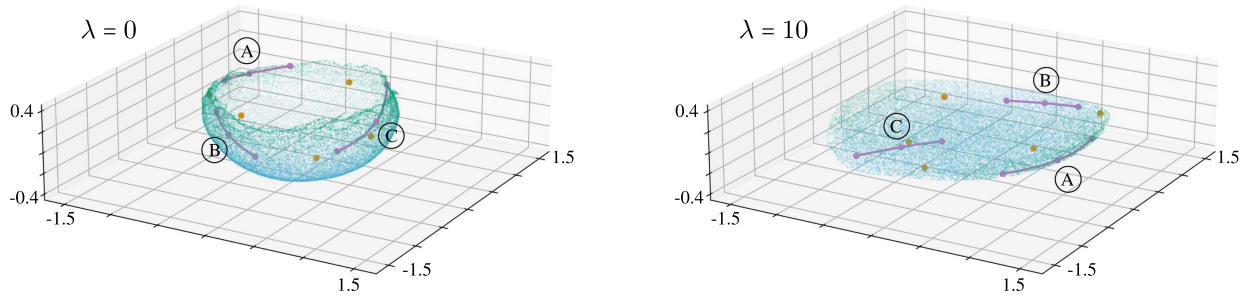

FIGURE 4. Comparison of the latent manifold $\phi(M) \subset \mathbb{R}^{16}$ for sundial dataset **(S)** for $\lambda = 0$ and $\lambda = 10$ and $\epsilon = \pi/8$ (colored points as in Fig. 1). The same training procedure as in Figure 1 was used. The three visualized components explain 97.8% ($\lambda = 0$) and 99.95% ($\lambda = 10$) of the variance. A threshold of 99% variance is reached for 6 ($\lambda = 0$) and 2 ($\lambda = 10$) components.

the flat torus than would be possible in $\mathbb{R}^3$. We observe that the encoder indeed makes use of that freedom for dataset **(R)**: The bottom right graph in Figure 5 shows the total amount of explained variance as a function of increasing subspace dimension. Apparently the embedding uses six Euclidean dimensions, even though $l = 5$ would be enough for a (potentially non-isometric) embedding ($SO(3) \cong \mathbb{R}P^3$, which embeds into $\mathbb{R}^5$, but not $\mathbb{R}^4$ [19, 22]). In Figure 6 we can similarly read off the used number of dimensions for dataset **(A)**. Here the situation is similar to, but not as obvious as before: to the best of our knowledge, it is not known whether there exists a smooth isometric embedding of the Klein bottle into less than five dimensions. However, isometric immersions (with self-intersections) exist into four dimensions (*cf.* [49]). Since our encoder loss functional cannot distinguish between immersions and embeddings (there is no term preventing self-intersections), the minimum dimensionality would thus be no larger than four, but our latent manifold actually uses five Euclidean dimensions in the case $\lambda > 0$. Interestingly, without bending regularization, *i.e.* for $\lambda = 0$, the embedding uses many more dimensions. As discussed above, this is not because a minimization of the isometry term would make this necessary. It is rather a manifestation of the strong degeneracy of the set of isometric embeddings, which are possible using arbitrarily many Euclidean dimensions. Figure 7 illustrates that the topology of the Klein bottle was reproduced for that dataset. It is well-known that no three-dimensional embedding without self-intersections exists. This explains the self-intersections in each of the visualized projections. In Figure 7 we also analyze the nature of those self-intersections occurring after projection: Denote the composition of ϕ with the projection onto a 3-dimensional subspace V by ϕ_V. Then such a self-intersection after projection is characterized by $\phi_V(x) = \phi_V(y)$ for $x \neq y$. Thus, a suitable indicator of a self-intersection at x is

$$\min_{y \in M, y \neq x} \frac{|\phi_V(x) - \phi_V(y)|}{d_M(x, y)}. \tag{4.3}$$

If this quantity is small, we expect a self-intersection at x. Recall that, when training the encoder separately, our method in principle does not prevent self-intersections. In Figure 6 we plot true distances on the Klein bottle against Euclidean distances of the corresponding embedded points in latent space. The resulting graph clearly indicates the absence of any self-intersections. While this might be due to the included decoder, which promotes injectivity of the embedding, we observed similar results when training the encoder separately. This plot further shows that, for $\lambda = 0$, the distances are more or less preserved up to the ϵ-value used in the training.Instead, for $\lambda = 1$, the isometry of the embedding is traded off for less bending.

4.4. Linear interpolation in latent space

Let x, y be two input images. We may obtain an interpolant between x and y in the following way. Let $\tilde{x}$ and $\tilde{y} \in \mathbb{R}^l$ be the latent codes corresponding to x, y and let $\tilde{z}$ be a linear interpolation of $\tilde{x}, \tilde{y}$. Decoding $\tilde{z}$ gives the

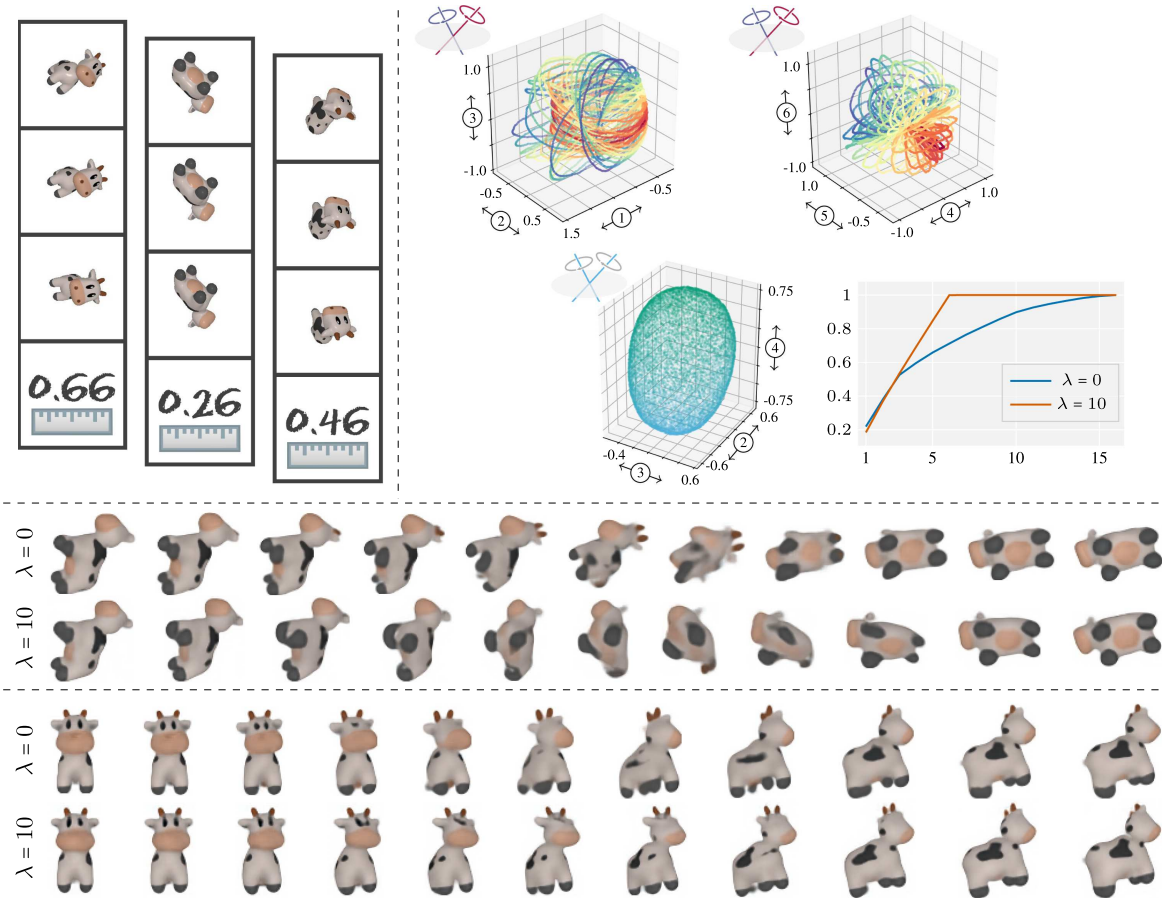

FIGURE 5. Visualization of the results for dataset (**R**) for $\lambda = 10$ and $\epsilon = \frac{\pi}{4}$ (with maximal distance being π). Encoder and decoder were trained separately. Training of the encoder was stopped after the value of the training loss did not decrease for 20 epochs. The decoder was trained until the reconstruction error R evaluated on a test set reached a threshold of 2×10^{-3}. Pairs with distance below $\frac{1}{20}$ of the maximal distance were rejected. The embedding is smooth, revealing the topology and geometry of the manifold $SO(3)$, and reasonable interpolations are obtained for $\lambda = 10$. On the left, three input triples and their distances are shown. To visualize the latent manifold, a PCA was performed on the point cloud in 16-dimensional latent space obtained by applying the embedding map ϕ to a large number of images from $M \cong SO(3)$. For the bottom right image we fixed a rotation angle and randomly sampled the rotation axis from S^2. For both other visualizations we regularly sample rotation axes from different latitudes of S^2 and then show points corresponding to rotations around each axis in the same color. The graph coordinates correspond to the principal components indicated by their labels. In components 2, 3, 4, a sphere-like structure can be observed, suggesting that these three components encode the rotation axis. The graph on the bottom right shows the the total amount of explained variance as a function of increasing subspace dimension, with a threshold of 99% reached for 6 dimensions for $\lambda = 10$. Below the dashed line we show examples of interpolations generated by interpolating linearly in latent space and subsequent decoding. For vanishing bending regularization $\lambda = 0$ those interpolations are unreliable, while they look very reasonable for $\lambda = 10$, even though the decoder was neither trained on linear interpolations in latent space nor was it regularized.

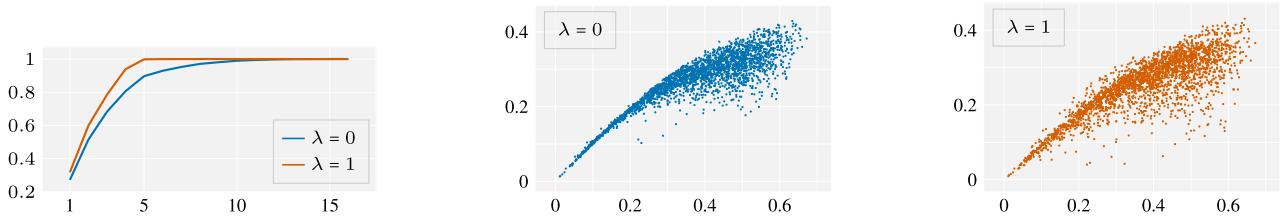

FIGURE 6. Comparison of two experiments for the dataset **(A)** for $\lambda = 0$ and $\lambda = 1$ and $\epsilon = \frac{1}{4\sqrt{2}}$ (with maximal distance being $\frac{1}{\sqrt{2}}$). Encoder and decoder were trained at the same time with $\kappa = 0.1$ and training was stopped once the reconstruction error R evaluated on a test set reached a threshold of 2×10^{-3}. Pairs with distance below $\frac{1}{20}$ of the maximal distance were rejected. The left graph shows the total amount of explained variance as a function of increasing subspace dimension, the total dimension of the latent space being 16. The threshold of 99% explained variance was reached for 5 and 11 dimensions for $\lambda = 1$ and $\lambda = 0$, respectively. The right graphs show the extrinsic latent space distance $|\phi(x) - \phi(y)|$ *versus* the intrinsic manifold distance $d_M(x, y)$ for random point pairs $(x, y) \in M$.

desired interpolant z between x and y. In formulas, $z = \psi(\alpha\phi(x) + (1-\alpha)\phi(y))$ for an interpolation coefficient $\alpha \in [0, 1]$. Flatness of the latent manifold is expected to improve the usefulness of interpolants obtained in this way across moderate distances, since one then expects the interpolant $\tilde{z}$ to be close to the latent manifold. This is qualitatively illustrated in Figure 5 for dataset **(R)**. For $\lambda = 0$, the decoder output of linear interpolations in latent space does not at all reproduce continuously rotating objects, whereas it clearly does for $\lambda = 10$. Since our focus is on the regularizing properties of our encoder loss functional $E^{\mathcal{S}_\epsilon}$, the decoder was intentionally not trained on linear interpolation and furthermore not additionally regularized. An additional training of the decoder on linear interpolants in latent space would likely improve the interpolation quality even more, and perhaps even allow the decoder to compensate for the deficiencies visible in Figures 5 and 10 for $\lambda = 0$. Due to the extrinsic curvature of the latent manifold, linear interpolation of course only makes sense for sufficiently close endpoints, as illustrated in Figure 10. To quantify the error of a linear interpolation in latent space as a function of interpolation distance, for a fixed test sample set $\mathcal{S} \subset M \times M$ we let $\mathcal{S}_\delta = \{(x, y) \in \mathcal{S} : d_M(x, y) \leq \delta\}$ and define

$$\mathrm{err}_{\mathrm{lat}}^2(\delta) = \frac{1}{|\mathcal{S}_\delta|} \sum_{x,y \in \mathcal{S}_\delta} |\phi(\mathrm{av}_M(x, y)) - \mathrm{av}_{\mathbb{R}^l}(\phi(x), \phi(y))|^2. \tag{4.4}$$

We further measure the L^2-error to the ground truth geodesic interpolation in image space using an error measure $\mathrm{err}_{\mathrm{im}}$ defined as

$$\mathrm{err}_{\mathrm{im}}(\delta)^2 = \frac{1}{|\mathcal{S}_\delta|} \sum_{(x,y) \in \mathcal{S}_\delta} \max\{\mathrm{err}_{\mathrm{i}}(x, y)^2 - \mathrm{err}_{\mathrm{b}}(x, y)^2, 0\} \text{ for} \tag{4.5}$$

$$\mathrm{err}_{\mathrm{i}}(x, y) = \| \mathrm{av}_M(x, y) - \psi(\mathrm{av}_{\mathbb{R}^l}(\phi(x), \phi(y)))\|_{L^2}, \tag{4.6}$$

$$\mathrm{err}_{\mathrm{b}}(x, y) = \| \mathrm{av}_M(x, y) - \psi(\phi(\mathrm{av}_M(x, y)))\|_{L^2}. \tag{4.7}$$

Here, $\mathrm{err}_{\mathrm{i}}$ is the error due to linear interpolation, and $\mathrm{err}_{\mathrm{b}}$ is the base reconstruction error, which occurs independently of interpolation. Note that it is expected (but not guaranteed) that $\mathrm{err}_{\mathrm{i}} \geq \mathrm{err}_{\mathrm{b}}$. For datasets **(A)** and **(R)**, Figure 8 compares $\mathrm{err}_{\mathrm{im}}(\delta)$ for different regularizations. The error is highest without encoder regularization, and it is in particular already high even for small distances. Our regularization reduces the error, particularly for small interpolation distances. In our controlled environment it is possible to also test the interplay of sampling radius and bending parameter λ.

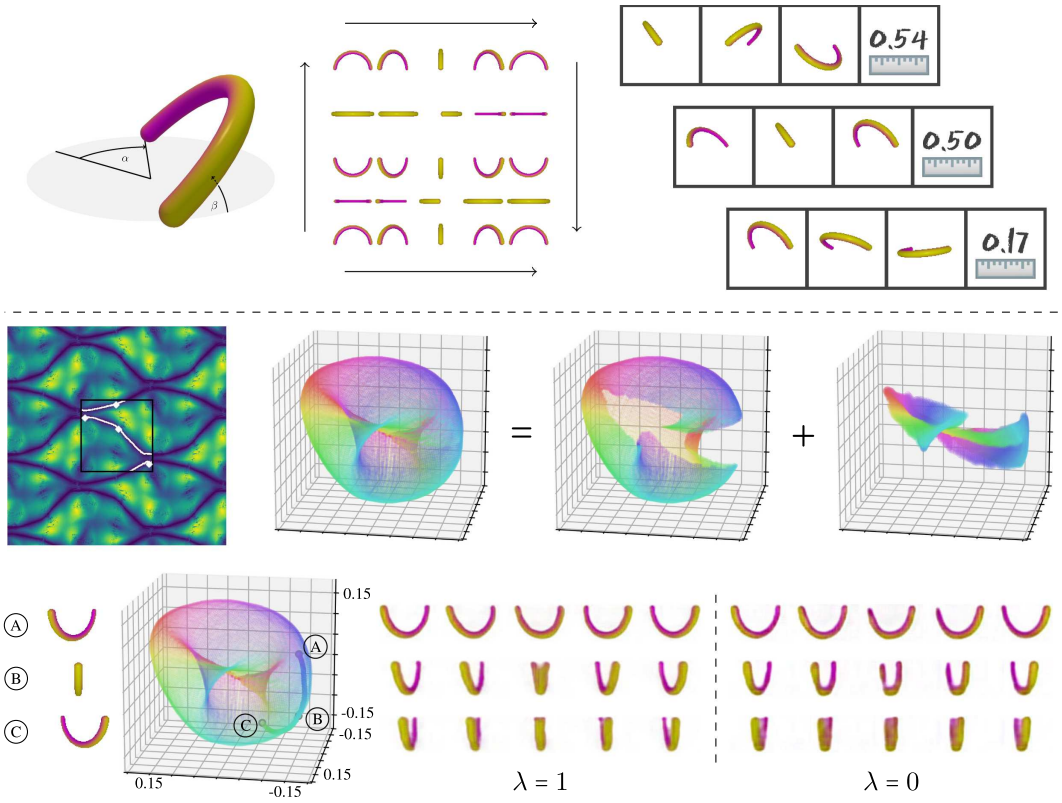

FIGURE 7. Visualization of the results for dataset (**A**) for $\lambda = 1$ (unless indicated otherwise). The same experiments as in Figure 6 were used. The first row illustrates how the data was generated: The first image shows how an arc is rotated by angles α and β. The second image shows variation of α in horizontal and variation of β in vertical direction, making visible the mirroring leading to the topology of the Klein bottle. On the right, three input triples and their distances are shown. The second row explains the latent manifold. First, a point cloud in 16-dimensional latent space is generated by applying the embedding map ϕ to a set of images obtained from a regular grid on $[0,1]^2$. This point cloud is then projected onto the first three components determined by a PCA. The first image visualizes the quantity (4.3) evaluated on this grid in the black frame. The image is extended in all directions according to the identification of the sides. The white lines were computed using the watershed algorithm. Each one of the lines (circles) is folded together, with the diamond markers showing the folding points. The folding becomes visible in the third image. Points are colored according to the original α coordinate, using a cyclic color map. The third row shows the embedding of a geodesic curve between A and C, where B is their geodesic midpoint, as well as the images obtained through linear interpolation in latent space for $\lambda = 0$ and $\lambda = 1$ using different starting points symmetric around B.

Recall that we use the same network architecture and penalty on the network weights for all ϵ, since we only consider a fixed range of ϵ values anyway. In Figure 9, we evaluated the interpolation errors $\mathrm{err}_{\mathrm{lat}}(\pi)$ and $\mathrm{err}_{\mathrm{im}}(\pi)$ from (4.4) and (4.5) for dataset (**S**) on a test set of arbitrarily chosen point pairs using sampling strategy (S3). For vanishing bending parameter λ we observe a dependence of the interpolation error on the sampling radius ϵ: A larger value of ϵ is associated with a smaller interpolation error, indicating that a large sampling radius ϵ also has some regularizing effect. A positive bending parameter λ more or less overrides the influence of the

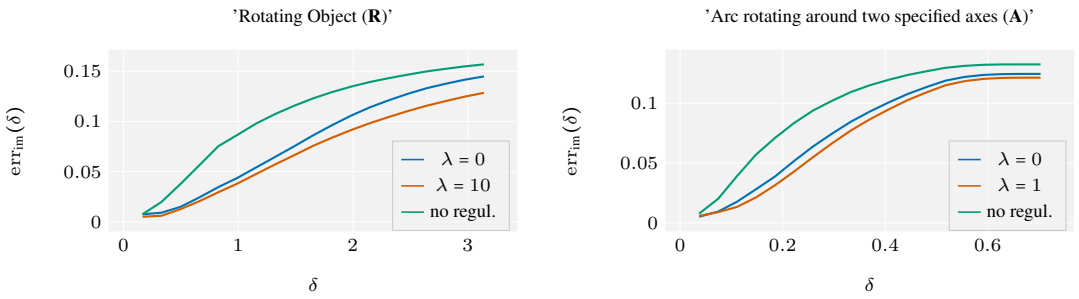

FIGURE 8. Comparison of the interpolation error for different regularizations for datasets **(R)** (left) and **(A)** (right).The same training procedures as stated in Figure 5 (for **(R)**) and Figure 6 (for **(A)**) were used.

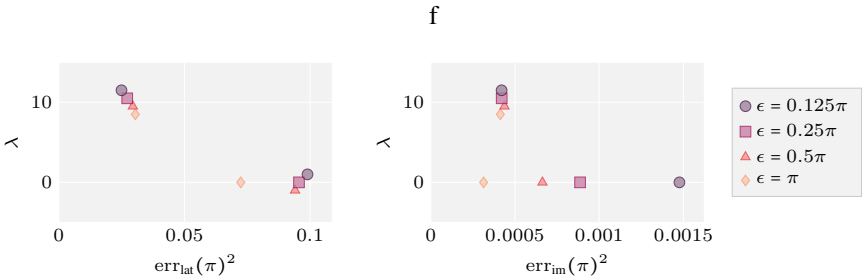

FIGURE 9. Average interpolation errors in latent and image space for training with different sampling radii ϵ computed across a test set consisting of random pairs with arbitrary distance on the upper hemisphere. The encoder was trained separately from the decoder, and the training was stopped after the value of the functional $E^{\mathcal{S}_\epsilon}[\phi]$ evaluated on a different test set did not decrease for 20 epochs. Pairs with distance below $\frac{1}{100}$ of the maximal distance were rejected. The decoder was trained until an accuracy of 10^{-5} was reached on the test set. This procedure was repeated three times with different random seeds. Note that the points were slightly shifted in y direction for better distinction.

sampling radius: The interpolation error gets consistently small, independent of ϵ. This confirms our hypothesis that the bending penalty produces latent manifolds with a smoother structure. This, in turn, is expected to reduce the generalization error in postprocessing tasks (indeed, the encoder was only trained on point pairs with distance below ϵ, but the interpolation errors $\mathrm{err}_{\mathrm{im}}$ and $\mathrm{err}_{\mathrm{lat}}$ are small for arbitrary point pairs).

4.5. Noise in the embedding due to image quantization

To illustrate the regularization properties of our loss functional, we did not add any artificial noise to our images, as noisy images would require additional tailored regularization. However, the process of generating images from low dimensional manifolds may introduce noise, for instance by discretization as pixel images in datasets **(A)** and **(R)**. We illustrate this effect on the dataset **(E)** by comparing smooth and binary images. Indeed, while the set of smooth ellipse images is infinite, the set of binary ellipse images is finite due to quantization. Thus, while the underlying manifold is still the same, the corresponding set in image space is different. For instance, the same pair of binary ellipse images may be sampled with different distances, leading to a type of noise. Figure 10 shows that the cylindrical structure of the resulting latent manifold $\phi(M)$ is not as clean in this case. Increasing the resolution of the images would reduce the disparity between the underlying manifold and the image manifold, leading to a reduction of the observed effect.

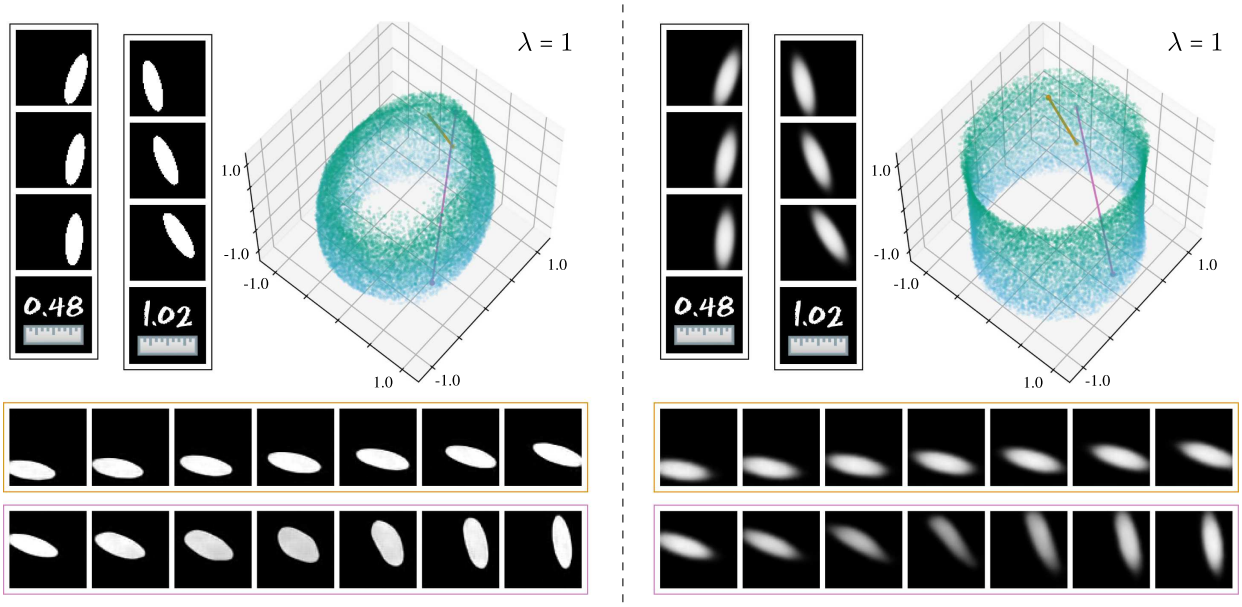

FIGURE 10. Visualization of the results for dataset (E) for $\lambda = 1$ and $\epsilon \approx 1.06$ (with maximal distance ≈ 4.32) for different ways of rendering ellipses: as an indicator function (left) and as smoothed indicator function with $k = 3$ (right). In each case, two input triples and their distances are shown. PCA coordinates 1, 2, 3 are used to visualize the latent manifold. On the bottom, example interpolations are visualized. Their corresponding interpolation path in latent space is shown in the PCA plot in the color of the frame. Encoder and decoder were trained separately. Training of the encoder was stopped after the value of the training loss did not decrease for 20 epochs. Pairs with distance below $\frac{1}{100}$ of the maximal distance were rejected. The decoder was trained until the reconstruction error R evaluated on a test set reached a threshold of 3×10^{-3} (binary images) and 10^{-4} (smoothed images).

5. Some open questions

We considered a two-step limit process for the discrete sampling loss functional $E^{\mathcal{S}_\epsilon}$, where we first let $|\mathcal{S}_\epsilon| \to \infty$ and then $\epsilon \to 0$. It seems plausible that one could also perform both limits simultaneously. Furthermore, we restricted the space of admissible functions, or rather admissible deep neural networks, according to ϵ. As discussed in Section 3.2, this restriction might, at least in certain settings, not be necessary. The limit energy $\mathcal{E}$ depends on second order derivatives of the encoder mapping and shows some resemblance to elastic energies of thin shells. By classical spectral arguments for discrete elliptic operators as in [48], explicit gradient descent methods for the minimization of such elastic energies come with severe time step restrictions of the order h^4, where h is the spatial discretization width. It is unclear if the structure of our regularization loss implies similar restrictions on the learning rate (the time step of the stochastic gradient descent) used when training the encoder. A simultaneous training of encoder and decoder *via* a minimization of the (weighted) sum of encoder regularization loss and reconstruction loss may have additional regularizing effects, since some embeddings may be more favorable for the decoder than others. Such effects have yet to be understood, as well as whether and how the decoder should eventually be regularized. This of course depends on the application in mind. For larger values of λ, the loss functional ensures that linear interpolation in latent space becomes a better approximation to Riemannian interpolation on the latent manifold. Correspondingly, we have seen a stronger flattening of the embedding in Figure 4. One might expect that this automatically improves the output of the

decoder applied to linear interpolations in latent space, as depicted in Figure 5. However, in the example of Figure 7, higher values of λ do not come with an improvement in the interpolation quality. This is also reflected quantitatively in Figure 8. In fact, for more complicated latent manifolds, various factors might additionally impact the interpolation error, such as the sign of the intrinsic curvature, an increased tangential distortion of the encoder map for flatter latent manifolds, or the interplay of the differently trained encoder and decoder map.

Acknowledgements. This work was supported by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) *via* project 211504053 - Collaborative Research Center 1060 and *via* Germany's Excellence Strategy project 390685813 - Hausdorff Center for Mathematics and project 390685587 - Mathematics Münster: Dynamics-Geometry-Structure.

Data availability statement. The code used in this paper is available online in a Gitlab repository `gitlab.com/jubrau/LBD` and in the datastore of the University of Münster `doi:10.17879/48978454901` [13].

References

[1] P.-A. Absil, R. Mahony and R.Sepulchre, Optimization algorithms on matrix manifolds. Princeton University Press (2009).
[2] R. Adams and J. Fournier, Sobolev Spaces. ISSN, Elsevier Science (2003).
[3] M. Atzmon, A. Gropp and Y. Lipman, Isometric autoencoders. Preprint: `arXiv:2006.09289` (2020).
[4] G. Aubert and P. Kornprobst, Can the nonlocal characterization of Sobolev spaces by Bourgain et al. be useful for solving variational problems? *SIAM J. Numer. Anal.* **47** (2009) 844–860.
[5] T. Aubin, Espaces de Sobolev sur les variétés riemanniennes. *Bull. Sci. Math.* **100** (1976) 149–173.
[6] M. Belkin and P. Niyogi, Laplacian eigenmaps and spectral techniques for embedding and clustering. *Adv. Neural Inf. Process.* **14** (2001).
[7] D. Berthelot, C. Raffel, A. Roy and I. Goodfellow, Understanding and improving interpolation in autoencoders *via* an adversarial regularizer. Preprint: `arXiv:1807.07543` (2018).
[8] H. Bolcskei, P. Grohs, G. Kutyniok and P. Petersen, Optimal approximation with sparsely connected deep neural networks. *SIAM J. Math. Data Sci.* **1** (2019) 8–45.
[9] R. Borghol, Some properties of Sobolev spaces. *Asymptot. Anal.* **51** (2007) 303–318.
[10] N. Boumal, An Introduction to Optimization on Smooth Manifolds. Cambridge University Press (2023).
[11] J. Bourgain, H. Brezis and P. Mironescu, Another look at Sobolev spaces. *Optimal Control and Partial Differential Equations A Volume in Honour of A. Benssoussan's 60th Birthday* (2001) 439–455.
[12] A. Braides, Γ-convergence for beginners, In Vol. 22 of *Oxford Lecture Series in Mathematics and its Applications*. Oxford University Press, Oxford (2002).
[13] J. Braunsmann, Python code for "Convergent autoencoder approximation of low bending and low distortion manifold embeddings". `https://doi.org/10.17879/48978454901` (2023).
[14] J. Braunsmann, M. Rajković, M. Rumpf and B. Wirth, Learning low bending and low distortion manifold embeddings. In *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops* (2021) 4416–4424.
[15] N. Chen, A. Klushyn, R. Kurle, X. Jiang, J. Bayer and P. Smagt, Metrics for deep generative models. In *Proc. of International Conference on Artificial Intelligence and Statistics*, PMLR (2018) 1540–1550.
[16] D.L. Donoho and C. Grimes, Image manifolds which are isometric to Euclidean space. *J. Math. Imaging Vis.* **23** (2005) 5–24.
[17] X. Feng and A. Prohl, Analysis of a fully discrete finite element method for the phase field and approximation of its sharp interface limits. *Math. Comput.* **73** (2004) 541–567.
[18] I. Gühring, G. Kutyniok and P. Petersen, Error bounds for approximations with deep ReLU neural networks in $W^{s,p}$ norms. *Anal. Appl.* **18** (2020) 803–859.
[19] W. Hantzsche, Einlagerung von Mannigfaltigkeiten in euklidische Räume. *Math. Z.* **43** (1938) 38–58.
[20] K. He, X. Zhang, S. Ren and J. Sun, Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proc. of IEEE International Conference on Computer Vision (ICCV)* (2015).
[21] E. Hebey, Sobolev Spaces on Riemannian Manifolds. Springer Berlin, Heidelberg (1996).
[22] H. Hopf, Systeme symmetrischer Bilinearformen und euklidische Modelle der projektiven Räume. *Vierteljschr. Naturforsch. Ges. Zürich* **85** Beiblatt (Festschrift Rudolf Fueter) (1940) 165–177.
[23] K. Hornik, M. Stinchcombe and H. White, Universal approximation of an unknown mapping and its derivatives using multilayer feedforward networks. *Neural Netw.* **3** (1990) 551–560.
[24] D.Q. Huynh, Metrics for 3d rotations: Comparison and analysis. *J. Math. Imaging Vis.* **35** (2009) 155–164.
[25] K. Kato, J. Zhou, T. Sasaki and A. Nakagawa, Rate-distortion optimization guided autoencoder for isometric embedding in Euclidean latent space. In *Proc. of the International Conference on Machine Learning*, PMLR (2020) 5166–5176.
[26] D.P. Kingma and M. Welling, Auto-encoding variational Bayes. Preprint: `arXiv:1312.6114` (2013).

[27] D.P. Kingma and J. Ba, Adam: A method for stochastic optimization, edited by Y. Bengio and Y. LeCun. In *Proc. of 3rd International Conference on Learning Representations (ICLR)* (2015).
[28] D. Kohli, A. Cloninger and G. Mishne, Ldle: low distortion local eigenmaps. *J. Mach. Learn. Res.* **22** (2021) 1–64.
[29] A. Kreuml and O. Mordhorst, Fractional sobolev norms and bv functions on manifolds. Preprint: arXiv:1312.6114 (2018).
[30] N.H. Kuiper, On C^1 isometric embeddings II. In Vol. 58 *Nederl. Akad. Wetensch. Proc. Ser. A* (1955) 683–689.
[31] N.H. Kuiper, On C^1-isometric imbeddings I. In Vol. 58 *Indagationes Mathematicae (Proceedings)*. Elsevier (1955) 545–556.
[32] J. Lellmann, K. Papafitsoros, C. Schönlieb and D. Spector, Analysis and application of a nonlocal Hessian. *SIAM J. Imaging Sci.* **8** (2015) 2161–2202.
[33] M. Loève, Probability Theory I. Springer New York, NY (1977).
[34] M. Mitrea and M. Taylor, Boundary layer methods for Lipschitz domains in Riemannian manifolds. *J. Funct. Anal.* **163** (1999) 181–251.
[35] U. Mosco, Convergence of convex sets and of solutions of variational inequalities. *Adv. Math.* **3** (1969) 510–585.
[36] J. Nash, C^1 isometric imbeddings. *Ann. Math.* (1954) 383–396.
[37] A. Oring, Z. Yakhini and Y. Hel-Or, Autoencoder image interpolation by shaping the latent space. Preprint: arXiv:2008.01487 (2020).
[38] G. Pai, R. Talmon, A. Bronstein and R. Kimmel, Dimal: Deep isometric manifold learning using sparse geodesic sampling. In *Proc. of the 2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, IEEE (2019) 819–828.
[39] E. Peterfreund, O. Lindenbaum, F. Dietrich, T. Bertalan, M. Gavish, I.G. Kevrekidis and R.R. Coifman, Local conformal autoencoder for standardized data coordinates. *Proc. of the National Academy of Sciences* **117** (2020) 30918–30927.
[40] P. Petersen, M. Raslan and F. Voigtlaender, Topological properties of the set of functions generated by neural networks of fixed size. *Found. Comput. Math.* **21** (2021) 375–444.
[41] G. Peyré and M. Cuturi, Computational optimal transport with applications to data science. *Found. Trends Mach. Learn.* **11** (2019) 355–607.
[42] M. Ranzato, C. Poultney, S. Chopra and Y. LeCun, Efficient learning of sparse representations with an energy-based model. *Adv. Neural Inf. Process.* **19** (2007) 1137.
[43] N. Ravi, J. Reizenstein, D. Novotny, T. Gordon, W.-Y. Lo, J. Johnson and G. Gkioxari, Accelerating 3d deep learning with PyTorch3d. Preprint arXiv:2007.08501 (2020).
[44] S. Rifai, P. Vincent, X. Muller, X. Glorot and Y. Bengio, Contractive auto-encoders: Explicit invariance during feature extraction. In *Proc. of the 28th International Conference on International Conference on Machine Learning* (2011).
[45] T. Sakai, Riemannian Geometry. Translations of mathematical monographs. American Mathematical Society, Providence, RI (1996).
[46] A. Schwartz and R. Talmon, Intrinsic isometric manifold learning with application to localization. *SIAM J. Imaging Sci.* **12** (2019) 1347–1391.
[47] H. Shao, A. Kumar and P. Thomas Fletcher, The Riemannian geometry of deep generative models. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition Workshops* (2018) 315–323.
[48] V. Thomée, Galerkin Finite Element Methods for Parabolic Problems, 2nd edition. In Vol. 25 of *Springer Series in Computational Mathematics*. Springer Berlin, Heidelberg, Berlin (2006).
[49] C. Tompkins, A flat Klein bottle isometrically embedded in Euclidean 4-space. *Bull. Am. Math. Soc.* **47** (1941) 508.
[50] A. Trouvé and L. Younes, Local geometry of deformable templates. *SIAM J. Math. Anal.* **37** (2005) 17–59.
[51] P. Vincent, H. Larochelle, Y. Bengio and P.-A. Manzagol, Extracting and composing robust features with denoising autoencoders. In *Proc. of the 25th international conference on Machine Learning* (2008) 1096–1103.
[52] H. Whitney, Collected Papers, Volume II, edited by E. James and T. Domingo (1992).
[53] L. Yonghyeon, S. Yoon, M. Son and F.C. Park, Regularized autoencoders for isometric representation learning. In *Proc. of International Conference on Learning Representations* (2021).
[54] L. Younes, Shapes and diffeomorphisms. In Vol. 171 of *Applied Mathematical Sciences*. Springer Berlin, Heidelberg (2010).
[55] C. Zhang, J. Liu, W. Chen, J. Shi, M. Yao, X. Yan, N. Xu and D. Chen, Unsupervised anomaly detection based on deep autoencoding and clustering. *Secur. Commun. Netw.* **2021** (2021).