ASME Journal of Mechanical Design
Online journal at:
https://asmedigitalcollection.asme.org/mechanicaldesign

Characterizing Designs Via Isometric Embeddings: Applications to Airfoil Inverse Design

Qiuyi Chen[1]
Department of Mechanical Engineering,
University of Maryland,
College Park, MD 20742
e-mail: qchen88@umd.edu

Mark Fuge
Department of Mechanical Engineering,
University of Maryland,
College Park, MD 20742
e-mail: fuge@umd.edu

Many design problems involve reasoning about points in high-dimensional space. A common strategy is to first embed these high-dimensional points into a low-dimensional latent space. We propose that a good embedding should be isometric—i.e., preserving the geodesic distance between points on the data manifold in the latent space. However, enforcing isometry is non-trivial for common neural embedding models such as autoencoders. Moreover, while theoretically appealing, it is unclear to what extent is enforcing isometry necessary for a given design analysis. This paper answers these questions by constructing an isometric embedding via an isometric autoencoder, which we employ to analyze an inverse airfoil design problem. Specifically, the paper describes how to train an isometric autoencoder and demonstrates its usefulness compared to non-isometric autoencoders on the UIUC airfoil dataset. Our ablation study illustrates that enforcing isometry is necessary for accurately discovering clusters through the latent space. We also show how isometric autoencoders can uncover pathologies in typical gradient-based shape optimization solvers through an analysis on the SU2-optimized airfoil dataset, wherein we find an over-reliance of the gradient solver on the angle of attack. Overall, this paper motivates the use of isometry constraints in neural embedding models, particularly in cases where researchers or designers intend to use distance-based analysis measures to analyze designs within the latent space. While this work focuses on airfoil design as an illustrative example, it applies to any domain where analyzing isometric design or data embeddings would be useful. [DOI: 10.1115/1.4063363]

Keywords: design optimization, design representation, design visualization, machine learning

1 Introduction

Analyzing past design data via machine learning has opened up new avenues for accelerating both human- and computer-generate designs in several ways [1]. For instance, in works like Refs. [2–7], researchers developed conditional inverse design models that can generate new designs satisfying the performance requirements, without going through time-consuming optimization. Other researchers [8–12] have focused on unconditional generation of designs, to either create more efficient shape parameterization functions or to augment an existing dataset with high-quality designs. Lastly, surrogate modeling methods have a long history using data-driven models to predict and optimize a design's performance so as to limit the need for computationally intensive first-principles solvers, such as finite element or finite volume solvers [13–17].

However, for all data-driven models researchers often need to ensure a dataset's quality by analyzing: how non-uniform it is; whether the data are concentrated, multi-modal, or biased over regions of data space; or whether the data points are noisy. Data analysis tools such as clustering or computing density or topological properties are typically used to characterize some of these factors, yet the curse of dimensionality [18] undermines their use in high-dimensional data space. Thus, practitioners usually first embed the data into some low-dimensional latent space via a dimension reduction method, and then conduct the analyses there instead [19–21]. But, given the large number of existing embedding methods, what properties do we need from the embedding to make such latent analyses reasonable? Chen et al. [22] proposed that for design data, we should care about the embedding's preservation of the geodesic distance, but that paper did not address the question of how to actually construct such an embedding.

This paper answers that question by proposing the recently developed *isometric autoencoder* [23–25] based on Riemann geometry to embed designs in a bidirectional and distance-preserving manner. We demonstrate the isometric embedding's necessity and practicality via a latent space analysis of the airfoil inverse design problem. Specifically, the paper provides the following contributions:

[1] Corresponding author.

Contributed by the Design Automation Committee of ASME for publication in the Journal of Mechanical Design. Manuscript received July 10, 2023; final manuscript received August 24, 2023; published online November 21, 2023. Assoc. Editor: Christopher Mccomb.

(1) We describe how to produce a bidirectional isometric representation with the isometric autoencoder. We apply this architecture to both some pedagogical toy problems and the real-world UIUC airfoil dataset[2]. These representations preserve the geodesic distance on the UIUC airfoil manifold in the form of the Euclidean distance in the latent space. We show how to use this isometric low-dimensional embedding as a proxy to investigate the quality of the UIUC dataset—i.e., its non-uniformity and multi-modality—through the lens of Hierarchical Density-Based Spatial Clustering of Applications with Noise (HDBSCAN) .

(2) We illustrate why preserving the geodesic distance when learning the design representation is necessary through a pedagogical counterexample incorporating flow-based models and optimal transport. We show that the lack of isometricity can lead to ambiguity when analyzing the UIUC airfoils' latent embedding.

(3) We use the isometric airfoil representation to study the quality and properties of the SU2 dataset of optimized airfoils. This dataset was produced by a gradient-based (i.e., adjoint) SU2 CFD solver and was used to train a CEBGAN (Condtiional Entropic Bézier Generative Adversarial Network) for inverse airfoil design. We unearth a pathology of the SU2 adjoint optimizer that it favors optimizing the angle of attack (AoA) more than the shape, and provides insights for future improvement. In addition, the isometric embedding sheds light on the robustness of different airfoils' to various flow conditions, which has implications for work in robust design optimization.

2 Background and Related Work

Before introducing the isometric autoencoder and how we use it for latent space cluster analysis, this section briefly reviews background and related work in clustering methods, how we define metric functions (in both the Euclidean and Geodesic sense), common latent space dimension reduction methods, and lastly basic definitions of isometry that we use through the rest of the paper.

2.1 Clustering. A design dataset can be regarded as a collection of designs sampled from a probability distribution supported by the set of valid designs. Among all the techniques that analyze the data samples to characterize their underlying distributions, clustering is a fundamental and popular one. It aims to assign the samples to different clusters by a certain algorithm on an unsupervised basis, insofar as the samples in the same cluster are more *similar* to each other than to those in other clusters. Owing to its ability to taxonomically describe the data distribution, clustering can highlight a distribution's non-uniformity and multi-modality, which makes it especially suitable to design datasets, as they are in general non-uniform and may contain several exemplary and timeless groups of designs that engineers desire to extract and imitate.

Out of all the existing clustering schemes, density-based clustering excels when the dataset consists of an unknown number of clusters of arbitrary shapes. Empirically [26,27], it is an ideal choice when the data distribution is supported by a set comprised of several separated-by-closed-neighborhoods components and the dataset is rich enough to delineate each, such that every disconnected component can be accurately identified as a cluster. Because of these properties, density-based clustering is ideal for cases where topological separation in data space among designs can indicate crucial design variation. There exist many density-based clustering models, among which DBSCAN [26] is probably the most celebrated one thanks to its efficacy withstanding the test of time [28,29], yet it is still not perfect [29,30] as a matter of course. Several successors like OPTICS [31], DENCLUE [32], and HDBSCAN [33] have since then being proposed to refine it. HDBSCAN [33] is a recent density-based hierarchical innovation with many great improvements, among which it in particular discards the annoying length scale hyperparameter ε of DBSCAN, hence we shall use this model in the later experiments for our convenience.

Despite clustering's usefulness, there is one critical aspect upon which any method's success hinges: the chosen distance function that describes similarity between points. Indeed, for clustering schemes either connectivity-based like hierarchical clustering [34], or centroid-based like k-means [35], or distribution-based like Gaussian mixture [36], or density-based like DBSCAN [26], selecting the distance function to quantify the resemblance between data samples is inevitably the first step and the foundation of the remaining process. As such, the distance function has outsized influence on the final result. Beyond clustering, this distance function is also important to any analyses wherein the data samples need to be juxtaposed to perceive their difference, such as in nearest neighbor search [37], Determinantal point processes (DPP) diversity quantification [38,39], etc., therefore it is worth being discussed in depth next.

2.2 Distance Functions. The canonical distance function (or metric) for an Euclidean space is the Euclidean distance. Despite being the most intuitive metric for low-dimensional spaces—thus usually being the default choice for clustering models—the Euclidean distance is frequently challenged in high-dimensional spaces. One well-known curse of dimensionality is the diminishing contrast between the maximum and the minimum L_p distance from a random query point to a series of random data points as the space dimensionality increases [40,41]. This cripples the effectiveness of distance-based algorithms like nearest neighbors or clustering on high-dimensional data. Apart from suffering the contrast-loss, Euclidean distance is also not even aware of the semantics of many high-dimensional data (i.e., what object(s) each data represents). For example, an image of object A may appear closer in Euclidean distance to an image of object B than to another image representing the same object A but in a different pose [42]. One plausible interpretation of this is that most high-dimensional data only reside on low-dimensional manifolds [43], yet a Euclidean distance defined over the ambient space (i.e., the Euclidean space containing that manifold) cannot take the shape and curvature of the embedded data manifold into account, as illustrated in Fig. 1.

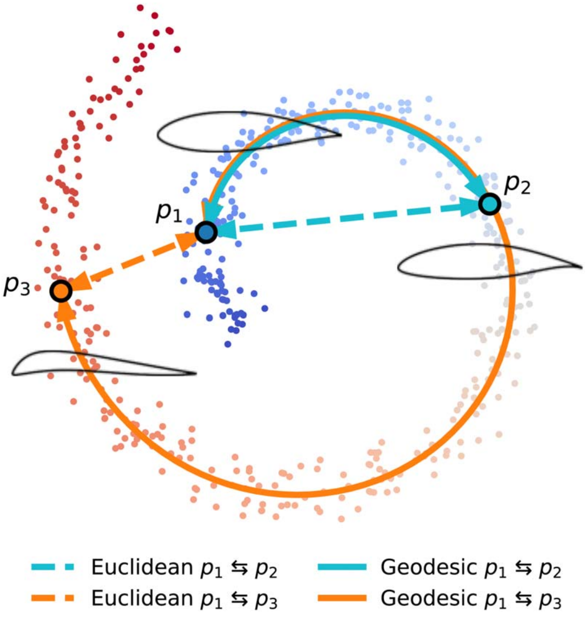

Fig. 1 Difference between Euclidean distance and geodesic distance when quantifying design differences

[2] https://m-selig.ae.illinois.edu/ads.html

Many alternative metrics have been proposed to overcome these issues. As an example, it is suggested that employing L_p distance functions of smaller (even fractional) p can mitigate the contrast-loss in high-dimensional spaces [41]. Nevertheless, just like their L_2 counterpart, these candidates are still unaware of the geometry of the data manifold, not to mention it is dubious to adopt them simply because of their better contrast, as L_p distance functions with $p < 1$ are not even well-defined, violating the triangular inequality that a legitimate metric should obey. Mahalanobis distance is another popular option [42,44,45], which is equivalent to performing a linear transformation over the entire ambient space prior to measuring distances with the Euclidean distance, so that when comparing points, different directions can be either emphasized or downplayed depending on their contribution to the semantics. However, such linear transformation needs to be trained beforehand by leveraging additional knowledge about the data's semantics, which is not always available. On top of that, linear transformation could be too primitive to simplify complicated nonlinear data manifold structure.

A probably more proper and natural choice of distance function for high-dimensional data is the geodesic distance (or Riemannian distance) [46,47] induced by the Euclidean metric over the data manifold, provided that the data manifold resides on a connected Riemann manifold—which can be intuitively called *extended* data manifold—such that the geodesics on it are well-defined. This distance function adapts to the geometry of the data manifold because it is defined as the length of the shortest path on it connecting two given points, thereby it is designed exclusively for that particular manifold and makes more sense both intuitively and empirically [48–52]. The greatest downside of this metric is its general lack of closed-form expression, since the evaluation of the geodesic length involves integration over the irregular manifold. Luckily, we can to some extent circumvent this issue by preserving this geodesic distance in a low-dimensional latent Euclidean space after establishing an *isometry* between the data manifold and a latent set in the latent space, so that the latent set serves as a proxy for that data manifold equipped with geodesic distance and we can equivalently perform geodesic-based analyses on it instead. To initiate this latent analysis, we need to start with dimension reduction to construct a map between the data space and the latent space at first.

2.3 Dimension Reduction. Dimension reduction aims to map data points in the high-dimensional space into a low-dimensional latent space while preserving some necessary information, such that the validity of certain analyses performed in the latent space can be ensured. For instance, t-SNE [20] retains the disconnectedness of the dataset [53], so that the disconnected subsets remain disconnected in the latent space. Isomap [54] preserves the graph distance on the neighborhood graph as an approximation of the true geodesic distance.

Generally, the data manifold is nonlinear, which often renders linear dimension reduction methods like principal component analysis (PCA) and non-negative matrix factorization (NMF) futile [54,55]. Out of all the nonlinear dimension reduction models, the autoencoder [56] is special for its simple reconstruction-loss-based formulation and the ability to not only map forward but also backward from the latent space to the data space. This ability to transform data bidirectionally is ideal for designs, as it can not only help us analyze designs conveniently in a low-dimensional setting —which is the focus of this paper—but also enable us to use its backward mode as a design parameterization to synthesize novel designs after we efficiently perform optimization or construct conditional generative models in the low-dimensional latent space. In addition, being a parametric model, once trained the autoencoder can be immediately applied to unseen new designs to derive the corresponding latent codes without starting from scratch. This advantage should compound as the dimensionality and scale of the design problem grows. It is therefore worthwhile to model that isometry for designs with the autoencoder rather than the other unidirectional, non-parametric methods like Isomap.

Compared with its counterparts like Isomap [54], LLE [55], UMAP [21], or diffusion map [57] that need to scrutinize each data point's neighborhood to infer the local manifold structure, the autoencoder has a more straightforward formulation which only needs us to globally minimize the reconstruction loss. However, it is still a dimension reduction method that can preserve a dataset's topological properties. This is because it approximately learns a *topological embedding*—which is a homeomorphism—between the dataset in the high-dimensional data space and latent set in the low-dimensional latent space, such that the original dataset's crucial topological properties like connectedness—which are invariant under homeomorphism—are preserved on the latent set [58]. Our claim about the autoencoder's homeomorphicity is bolstered by the rationale that both the encoder e and the decoder g are modeled by *continuous* neural networks, and the minimization of the reconstruction loss $\mathbb{E}_{x\sim\mathcal{X}}\|g \circ e(x) - x\|$ encourages the composition $g \circ e$ to be an identity function over the dataset $\mathcal{X}$ and drives both g and e to be *bijective* between the dataset $\mathcal{X}$ and the latent set $e(\mathcal{X})$ [59], which is the very definition of a topological embedding [58].

However, preserving the topological properties alone is not enough for clustering on the latent set, since there exists an infinite number of latent sets that each has different pairwise distances but is still homeomorphic to the same dataset (this non-uniqueness of homeomorphism is why flow-based models [60–62] can approximate different distributions with a fixed latent distribution, as we will see in Sec. 4.1.3). This may in consequence induce an infinite number of results for latent clustering and lead to ambiguity. Therefore, preserving a selected distance function in the data space is necessary, and this brings us to the preservation of the aforementioned geodesic distance on the data manifold.

2.4 Isometry. An *isometry* between two Riemann manifolds is a *locally isometric* diffeomorphism (i.e., a smooth homeomorphism), which preserves the geodesic distances between points. For simplicity, if we restrict this generic definition to our special case where this map is modeled by an autoencoder, then based on the discussions in Refs. [23–25], we say the autoencoder establishes an isometry between a Riemann data manifold in the high-dimensional data space and a Riemann latent manifold in the low-dimensional latent space—provided that they are *connected* sets and inherit their Riemann metrics from their Euclidean ambient spaces respectively—if the autoencoder is

(1) A diffeomorphism: Both the encoder and the decoder are modeled by *smooth* neural networks, and the autoencoder's reconstruction loss is also minimized to near zero over the Riemann data manifold, creating a *homeomorphism*.
(2) Locally isometric: All singular values of the Jacobians of both the decoder and the encoder need to be 1 at every point over the Riemann manifolds. Intuitively, this means the local linear transformation (i.e., the differential) does not stretch or compress the input along any direction tangent to the manifold.

On top of that, ideally we hope the latent space's dimension is equal to the latent manifold's dimension and the latent manifold is a convex set—which aligns its geodesic distance with the Euclidean distance—such that the geodesic distance between any pair of points on the data manifold equals the Euclidean distance between their corresponding latent codes. However, this ideal setting is not always encountered for the following reasons:

(1) The dataset (and thus the latent set) may not be connected. Remember this is what motivates us to use HDBSCAN to investigate the dataset's topology. Intuitively, this means there exist some intervals between these components that are not regularized for the autoencoder, such that while the distance within each component is preserved, these components can be arbitrarily close to each other in the latent space.

(2) The latent space's dimension could be larger than the dataset's dimension. There are many causes for this issue. For example, the dataset may not be a manifold, as it could be not locally Euclidean somewhere. The dataset could also be a manifold unable to be embedded in a space of the same dimension, as will be discussed later. Moreover, the dimension of the latent space needs to be determined beforehand, but we may not be able to estimate it accurately.

(3) The latent set is not necessarily convex, as its shape is determined by the shape of the dataset through the isometry. In addition, if the above dimension misalignment exists, the nonlinearity of the latent set may also destroy its convexity.

All these issues suggest that we need to somehow extend the dataset into a well-behaved connected Riemann manifold to establish that isometry robustly. We introduce methods to do this next.

3 Methodology

This section covers the two primary methods that we use in this paper's later experiments: (1) the isometric autoencoder and (2) a method for estimating the intrinsic dimension of the data manifold.

3.1 Isometric Autoencoder. Since the reconstruction loss enforces homeomorphicity, and both the decoder and the encoder are smooth models, to enforce isometry within an autoencoder we only need to make the encoder and the decoder locally isometric. We can accomplish this by having their Jacobians' singular values all equal to one over the dataset (or equivalently the latent set). This is easier said than done, due to the substantial computational cost entailed in explicitly deriving the Jacobian's singular values, particularly in high-dimensional cases. Thus, instead of regularizing those singular values explicitly, researchers typically impose the isometry constraint implicitly via random vectors sampled uniformly from a unit sphere $\mathbb{S}^{m-1}$ embedded in the m-D latent space, such that $\|J_g v\| = 1$ and $\|v^\top J_e\| = 1$ for every $v \in \mathbb{S}^{m-1}$, where J_g and J_e are the Jacobians of the decoder and the encoder respectively. It can be verified that this leads to unitary singular values [24]. More importantly, we can compute the Jacobian-vector product (JVP) or vector-Jacobian product (VJP) far more efficiently than the full Jacobian via modern packages like functorch and JAX, which can also invoke the optimal auto-differentiation mode (forward mode for JVP and backward mode for VJP) providing efficient backpropagation.

Enforcing isometry only over the dataset may not be enough, due to the three problems mentioned in Sec. 2.4. To overcome these issues, we employ the latent interpolation or mixup method [23,63] to additionally enforce isometry over some interpolated latent points. Specifically, these interpolated points are sampled uniformly from the lines connecting some random pairs of real latent points (i.e., the corresponding latent points of real data samples). This latent interpolation approximately samples points from the *convex hull* of all real latent points, so that by enforcing isometry over this extended convex latent set, we equivalently enforce isometry over its image in the data space, which is a connected Riemann manifold covering the dataset. This connected manifold can thus be regarded as an extended data manifold that has a well-defined geodesic distance and a convex latent set. To make sure the extended latent set is homeomorphic to the extended data manifold, we additionally minimize the cycle consistency loss [64] over the interpolated latent points.

Altogether, the loss function of the isometric autoencoder consists of four square terms:

$$L = \mathbb{E}_{x\sim\mathcal{X},\, z\sim\overline{e(\mathcal{X})},\, v\sim\mathbb{S}^{m-1}}\{[g\circ e(x) - x]^2 + \beta[e\circ g(z) - z]^2 + \lambda[\|J_g(z)\cdot v\| - 1]^2 + \lambda[\|v^\top\cdot J_e(x)\| - 1]^2\} \tag{1}$$

where β and λ are weight coefficients for cycle consistency loss and isometry regularization respectively, $\overline{e(\mathcal{X})}$ is the extended latent set consisting of the interpolated points, and the random vectors (x and v) are sampled uniformly from their respective sets. To sample v uniformly from $\mathbb{S}^{m-1}$, we can take advantage of the radial symmetry of the unit Gaussian distribution by simply normalizing the vectors sampled from it.

3.2 Intrinsic Dimension Estimation. To construct the autoencoder, we must first determine the latent space dimension (m). This dimension is vital for two reasons. On one hand, if the dataset constitutes a manifold—which is locally Euclidean of certain dimension—and we attempt to fit an autoencoder whose latent space is lower than this intrinsic dimension, the autoencoder will never establish a homeomorphism between the dataset and the latent set due to the topological invariance of dimension [65]. This would cause the autoencoder to map different data points to the same latent point, making the encoder no longer injective. This "data collapse" is detrimental to latent clustering for the following reasons. First, the autoencoder is thereby no longer isometric, which makes the intrinsic latent space metric unreliable for clustering. In addition, although the connectedness of the dataset is still preserved on the latent set thanks to the encoder's continuity, the disconnectedness is not guaranteed to remain; this means that multiple disconnected clusters in the data space may be merged into a single connected one in the latent space. This merging would clearly mislead any clustering algorithm. While some may argue that this collapse is not always unfortunate since some trivial data dimensions might be eliminated in our favor in the latent space, this removal is out of our control and it is unwise to rely on luck. On the other hand, avoiding this collapse problem by setting the latent space dimension higher than absolutely necessary creates its own problems: our whole reason for conducting dimension reduction in the first place is to reduce clustering problems that occur in high-dimensional spaces and make isometry regularization more efficient.

To this end, we employ the maximum likelihood estimation (MLE) [66] with bias correction [67] as the intrinsic dimension estimator, which performed well in our previous work [68]. In brief, MLE presumes locally constant data density and Poisson-distributed number of neighbors around each point. Under that model, the likelihood maximization leads to the local estimate

$$\hat{m}_k(x) = \left[\frac{1}{k-1}\sum_{j=1}^{k-1}\log\frac{T_k(x)}{T_j(x)}\right]^{-1} \tag{2}$$

where k is the pre-selected number of neighbors for each evaluation and $T_j(x)$ is the Euclidean distance between x and its jth nearest neighbor. The debiased global estimator [67] summarizing these local results is

$$\bar{m}_k = \left[\frac{1}{n}\sum_{i=1}^{n}\hat{m}_k(x_i)^{-1}\right]^{-1} = \left[\frac{1}{n(k-1)}\sum_{i=1}^{n}\sum_{j=1}^{k-1}\log\frac{T_k(x_i)}{T_j(x_i)}\right]^{-1} \tag{3}$$

More details about improving its precision can be found in our previous work [68].

Even when we estimate the data manifold's dimension d accurately, it is not necessarily the proper number for the latent dimension, because we cannot always embed some manifolds of a given dimension into an ambient Euclidean space of the same dimension. The Klein bottle is a well-known example—this two-manifold can only embed successfully in at-least-4D Euclidean spaces. Luckily, the Whitney embedding theorem [65] suggests that to obtain a homeomorphic autoencoder we can upper bound the latent dimension by $2d$. Therefore, a practical way to attain the optimal latent dimension is to incrementally increase it from d all the way to $2d$ until the final reconstruction loss becomes marginal. For data manifolds that do not have complexities or pathologies similar to the Klein bottle example,

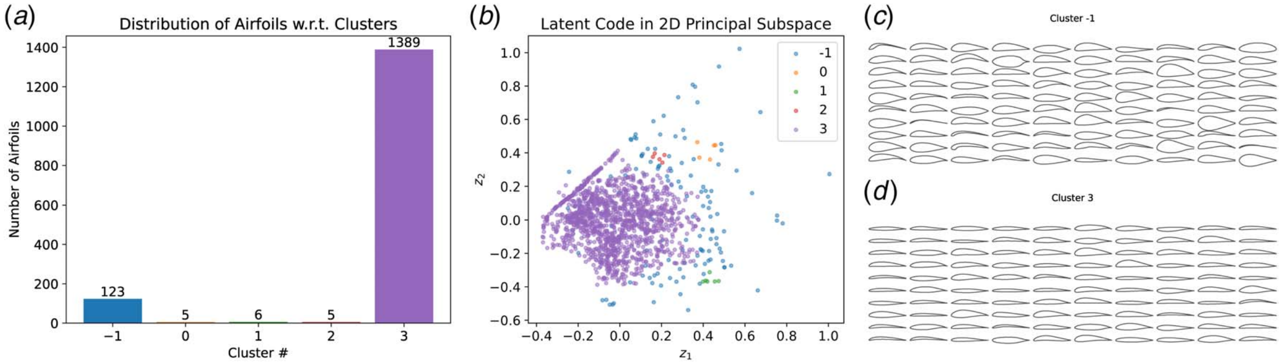

Fig. 2 Result of latent HDBSCAN on UIUC airfoil dataset, with $m_{pts} = m_{clSize} = 5$. Here (*a*) shows the amount of airfoils assigned to each cluster by HDBSCAN and (*b*) illustrates the spatial distribution of the airfoils in each cluster in the 2D space spanned by the top-2 principal components of the latent codes, with each point marked according to its corresponding cluster in (*a*). All plots hereafter, unless otherwise specified, have the same configuration: (*a*) airfoil distribution (#-1 for noise), (*b*) PCA of isometric latent set, (*c*) airfoils regarded as noise (#-1), and (*d*) airfoils in cluster #3.

we would expect the result to be much closer to d than $2d$, and thus attainable in a few trial-and-error shots.

4 Experiments and Discussion

In this section, we use the airfoil designs as a concrete example to demonstrate how the isometric autoencoder can help engineers analyze high-dimensional designs intuitively in a low-dimensional setting for different purposes. Overall, we first apply the isometric autoencoder to the UIUC airfoil dataset to obtain a low-dimensional isometric latent representation of the historical airfoil designs. Then, we perform latent HDBSCAN to locate common past airfoil designs—for pedagogical effect, we will disrupt the isometricity of this autoencoder in one experiment to highlight the importance of preserving distance for latent clustering. In the second half of our experiments, we introduce and employ the resulting isometric autoencoder to derive the latent representation of a subset of airfoils optimized by the SU2 suite under a large variety of boundary conditions. We then use this isometric embedding to investigate how the airfoil shape and angle of attack are related to different input conditions.

4.1 UIUC Airfoil Latent Clustering

4.1.1 Isometric Representation of UIUC Airfoils. To help generate smooth airfoil curves, we integrate the Bézier layer used in our previous work [11] into the decoder as its output layer. Our decoder and encoder also inherit their general architectures respectively from the generator and discriminator in Ref. [11], since their complexity was previously sufficient for this dataset. We set both β and λ to 0.01 for the isometry regularization. We set the latent space dimension to three, based on prior published experiments with the UIUC dataset [68]. After constructing the autoencoder, we then train it with an ADAM optimizer of learning rate 0.0001 for 6000 epochs with batch size 32 over the UIUC dataset consisting of 1528 airfoils. During each epoch, the autoencoder is trained 48 times over different shuffled mini-batches.

After training, the autoencoder achieves a reconstruction error of approximately 5×10^{-5} and a cycle consistency loss around 9×10^{-4}, while the isometry regularization errors of the decoder and the encoder are around 8×10^{-5} and 4.5×10^{-3} respectively. These results indicate that the isometric autoencoder does not sacrifice reconstruction error relative to the unregularized version. We then use the isometric encoder to map all of the UIUC airfoils into the latent space, and we use these encoded coordinates for the following latent analyses. Figure 2(*b*) shows the 2D principal projection of this (3D) latent set.

4.1.2 Latent Clustering With HDBSCAN Using the Isometric Autoencoder. At first, we perform HDBSCAN on the latent set with both m_{pts} and m_{clSize} set to 5—namely the minimum number of neighbors a given point should have to qualify as a core and the minimum number of cores that a cluster should have. This HDBSCAN configuration classifies <10% of the latent points as noise and does not produce as many trivial tiny clusters, thus reasonably grasping the overall structure of the dataset. Figure 2 shows that over 90% of airfoils in the UIUC dataset are considered density-connected by HDBSCAN under this hyperparameter setting and assigned to cluster #3, while there are only a few (139) outliers left surrounding it (Fig. 2(*b*)). In other words, in UIUC dataset there exists a dominant connected subset in which the airfoils are densely distributed everywhere (otherwise they would not become cores in HDBSCAN). This cluster can be regarded as the group of "canonical" airfoil designs. The illustrations in Figs. 2(*c*) and 2(*d*) seem to testify to this claim, as subjectively those noise airfoils have a higher variety of unusual shapes than those in cluster #3, although this is a qualitative conjecture on our part.

Despite its mechanical and systematic procedure, HDBSCAN is still highly sensitive to varying m_{pts} and m_{clSize}, which can significantly alter each point's cluster assignments. For instance, when we increase m_{pts} and m_{clSize} to 10 and 50 respectively to sift out low density regions and small clusters more radically, we can get a different result as shown in Fig. 3. Due to the current higher standard for what constitutes a cluster, only a few (four) regions of higher density in the previous cluster #3 now qualify as clusters. In that sense, we can regard these four as the most typical groups of airfoils among all canonical airfoil designs. We can plot the mass centers of these clusters as the representatives of them, as shown in Fig. 3(*c*).

4.1.3 What Happens to Latent Clustering When We Destroy Isometry? It may not be easy to appreciate the importance of isometry regularization without a contrast, so here we purposefully sabotage the airfoil autoencoder's isometricity to demonstrate how its absence may lead to distortion of the latent set and hence to a misleading clustering result. Specifically, our overall "trick" below will be to enforce the exactly same autoencoder reconstruction loss, but selectively destroy the latent space's isometricity using a tunable bijective distortion that allows us to increasingly "break" only the isometricity between the design and latent spaces.

To do this in practice, we first unlock the autoencoder's isometricity by attaching a flow-based model like RealNVP [61] to its latent space. Specifically, let g and e be the decoder and encoder of the isometric airfoil autoencoder and f be the RealNVP, then construct a new latent set with $f \circ e(\mathcal{X})$, where $\mathcal{X}$ denotes the UIUC dataset. We can thereby regard $e' := f \circ e$ and $g' := g \circ f^{-1}$ as the new pair of encoder and decoder between the UIUC dataset and the new latent set, where f^{-1} can be readily retrieved given that f is a

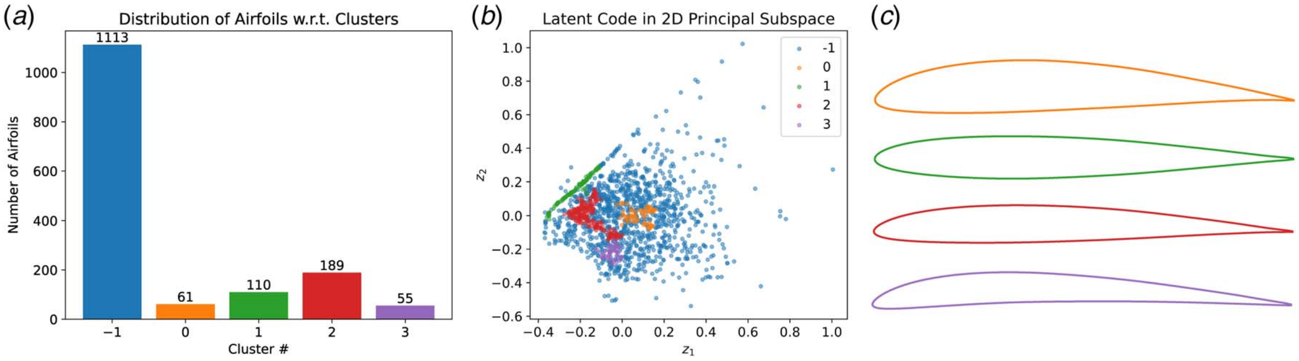

Fig. 3 Result of latent HDBSCAN on UIUC airfoil dataset, with $m_{pts}=10$ and $m_{clSize}=50$: (*a*) airfoil distribution (#-1 for noise), (*b*) PCA of isometric latent set, and (*c*) airfoil representatives

diffeomorphism between the old and new latent spaces by construction. Therefore, the reconstruction loss does not change at all for any flow f, as $\|g' \circ e'(x) - x\| = \|g \circ f^{-1} \circ f \circ e - x\| = \|g \circ e(x) - x\|$. In other words, through this new autoencoder we get a new latent set that is also homeomorphic to the dataset (and thus to the old isometric latent set), but we can train f while fixing e and g to tamper with its isometricity.

Next we show this unlocked autoencoder can obtain a latent set (as shown in Fig. 4) dramatically different from the isometric one in Fig. 2(*b*). We start by constructing a 3D Gaussian mixture target distribution $p_t(z)$ of three components centered at [1, 0, 0], [0, 1, 0], and [0, 0, 1] respectively, each having an isotropic standard deviation equal to 0.1. Then we drive the empirical distribution $p_f(z)$ of the new latent codes $e'(\mathcal{X})$ to p_t by only training f and leaving g and e fixed. Since we do not know the probability density function over the isometric latent set $e(\mathcal{X})$, there is no way to train f via log-likelihood maximization, so we achieve this instead by minimizing the Sinkhorn divergence between p_t and p_f. More information about this method can be found in Ref. [69]. This encourages f to transform the old isometric latent set into a new one with most of its points concentrating around [1, 0, 0], [0, 1, 0], and [0, 0, 1], and thus may form three big clusters instead of one. If our claim is true that the lack of isometricity can lead to distortion, this three-cluster latent set should be attainable as long as f has enough complexity.

Figure 4 shows the clustering result of the non-isometric latent set after training. It now consists of three major clusters (#7, #8, #13) corresponding to the three Gaussian components of p_t. This stands in stark contrast to the single giant cluster in Fig. 2(*b*), despite the two models having identical reconstruction error, and exemplifies the ambiguity problem mentioned in Sec. 2.3. In other words, we cannot use reconstruction error alone to know that the resulting latent space distances preserve isometry, and the isometric autoencoder provides a mean to regularize this directly. While in real applications, an autoencoder may not distort the dataset's distance and topology quite as severely as in our extreme example, naïve models have no protection against it. As such, it is therefore always worthwhile to switch on the isometry regularization when doing latent analyses relying on distance functions.

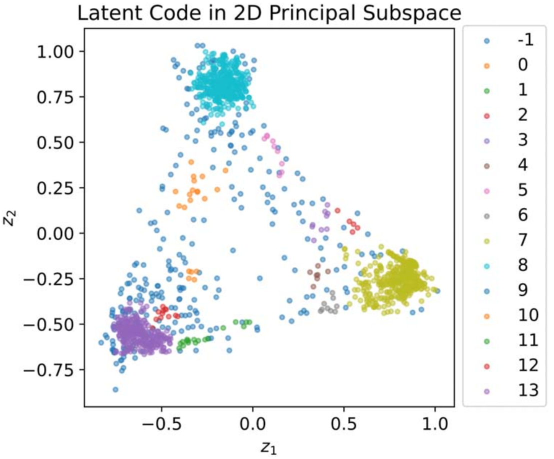

Fig. 4 PCA of and HDBSCAN on non-isometric latent set, with $m_{pts}=m_{clSize}=5$

4.2 Latent Shape Analysis of SU2 Airfoil Optimization.

As mentioned earlier in Sec. 2.3, one great advantage of the autoencoder over many other non-parameterized dimension reduction methods is its ability to perform *amortized* inference, i.e., it can immediately process unseen data samples without retraining. In the following experiments, we exploit this advantage to analyze the SU2 airfoil dataset [5] consisting of 1245 airfoil-AoA pairs that are optimized under a variety of boundary conditions by the SU2 CFD toolset. Specifically, we postulate that these optimized airfoil shapes still reside on the UIUC airfoil manifold, and apply the pre-trained UIUC airfoil autoencoder directly on the SU2 dataset (shapes only, without AoAs) to derive its isometric latent representation for the following latent analyses.

4.2.1 Latent Clustering of Optimized Airfoils. We perform HDBSCAN with $m_{pts}=10$ and $m_{clSize}=30$ on the latent set of SU2 airfoils first and illustrate its result in Fig. 5. Compared with the 1528 UIUC airfoils that cover a large area and comprise a giant cluster, the 1245 SU2 airfoils only occupy a few small regions and form five clusters, as we can see in Fig. 5(*a*). This higher regional density is the prime reason why in this case we do not reuse the previous $m_{pts}=m_{clSize}=5$ setting for Fig. 2, as otherwise it will produce over 30 tiny clusters and leave about 25% airfoils categorized as noise, which is not reasonable.

The SU2 airfoils' distinctive regional concentration probably stems from the way they were created. Chen et al. [5] observed that the gradient-based airfoil adjoint optimization often arrived at sub-optimal local optima. Consequently, for each input condition they performed eight restarts of the adjoint optimization by selecting a diverse set of starting airfoils sampled from BézierGAN approximating the UIUC airfoil distribution. In Ref. [5], the final training set included only the highest efficiency final design from these eight optimization trials. In practice, in many cases the SU2 optimizer found that altering only the AoA was sufficient to find the most efficiency design, compared to modifying the airfoil shape. As such, we would expect to find dense clusters of initial shapes (in the latent space) for the cases where the optimized design only modified the AoA. We shall verify if this is true next.

4.2.2 How Much Do Airfoils Morph in Optimization? To investigate the airfoil shape's degree of variation during

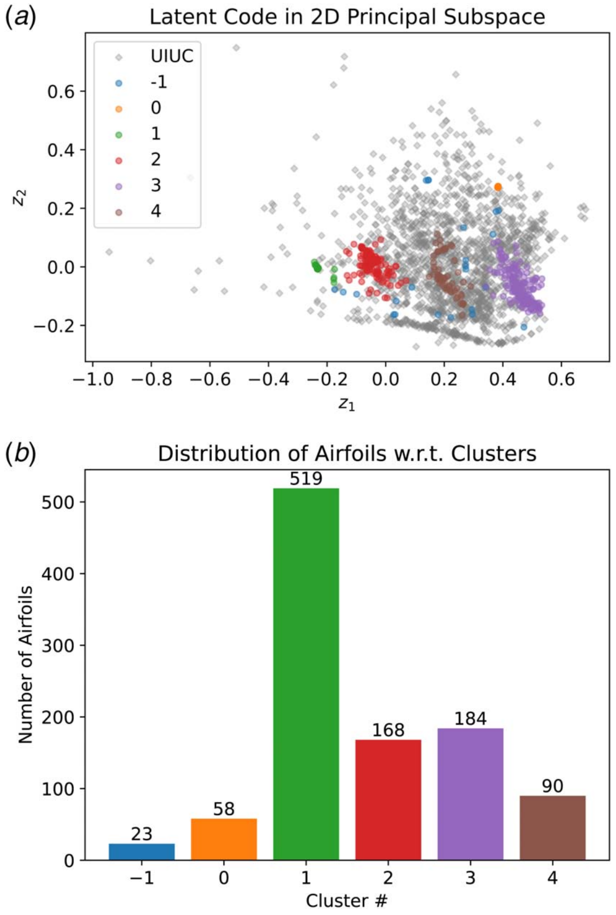

Fig. 5 Result of latent HDBSCAN on SU2 airfoil dataset, with $m_{pts} = 10$ and $m_{clSize} = 30$: (*a*) PCA of SU2 dataset's isometric latent set and (*b*) SU2 airfoil distribution (#-1 for noise)

optimization, especially in comparison to that of the AoA, we perform PCA on the Cartesian product of the normalized 3D latent code and normalized AoA—namely their 4D concatenation. To avoid distortion of the latent code, when normalizing/standardizing the code we use the standard deviation across all three latent dimensions as the scaling factor, instead of scaling it dimension-wisely after mean centering. By doing this, we can make sure the normalized latent set is still isometric to the dataset *up to a scale factor* [23], so that some potentially trivial latent dimensions will not be emphasized relative to the principal ones after normalization.

The new PCA result is plotted in Fig. 6, where each point is marked according to the cluster found earlier plotted in Fig. 5. In contrast to Fig. 5(*a*), the introduction of AoA in PCA induces a reorientation that reveals each cluster's linear pattern along the AoA direction (which is illustrated by the dashed line in Fig. 6). There are two naïve takeaways from this graph, if we (for now) ignore some technical caveats and presume that smaller variation in the direction perpendicular to the AoA direction indicates smaller change in shape:

(1) In general, the SU2 adjoint optimizer tends to optimize the airfoil's lift-drag efficiency for different input conditions—Re, Ma, and lift coefficient—more by adjusting its AoA than by morphing its shape, given that each cluster's variation along the AoA direction is more substantial than that along the perpendicular direction.

(2) Not all airfoils are created equal. We can notice that the airfoil shapes in clusters #0 and #1 have much smaller variation compared with the ones in clusters #2, #3, and #4. If it is true that almost all airfoils in each cluster are optimized from the same initial design from among the provided eight restarts,[3] then this suggests some of the eight initial airfoils are better starting shapes (i.e., lie close to the basin of the optima) under certain ranges of boundary conditions compared to others, leading the SU2 optimizer to only need to adjust the AoA to improve efficiency. The overwhelming number of airfoils in cluster #1 (Fig. 5(*b*)) also reflects this inequality, suggesting that the initial design that, when optimized, yielded the shapes in cluster #1 is not just optimal for some conditions, but rather broadly

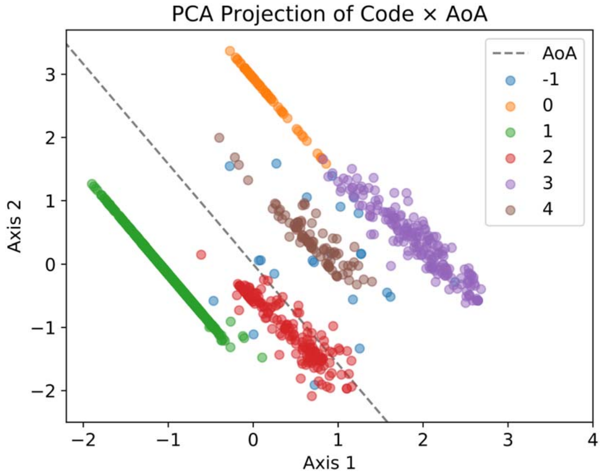

Fig. 6 2D principal projection of the Cartesian product of airfoil latent code × AoA

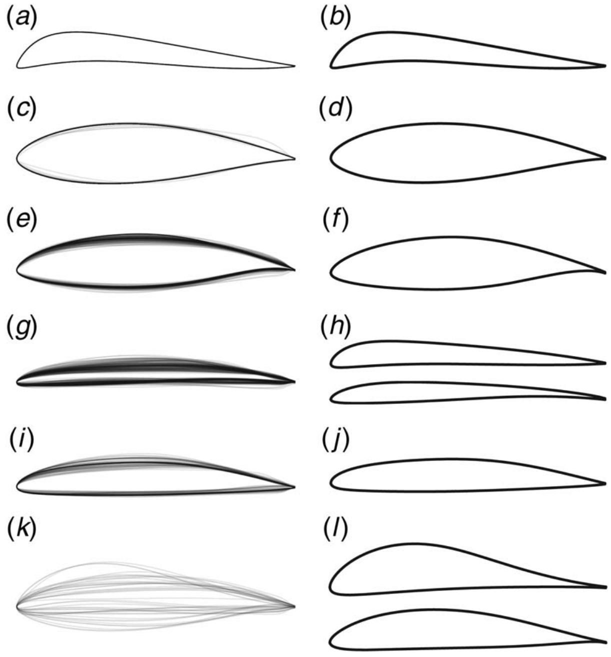

Fig. 7 Airfoils in different clusters and their initial designs: (*a*) cluster #0, (*b*) initial design #0, (*c*) cluster #1, (*d*) initial design #1, (*e*) cluster #2, (*f*) initial design #2, (*g*) cluster #3, (*h*) initial designs #3, (*i*) cluster #4, (*j*) initial design #4, (*k*) cluster #-1 (*Noise*), and (*l*) remaining initial designs

[3] This is likely considering how separated these clusters are away from one another in the latent space. We will also see this visually later in Fig. 7.

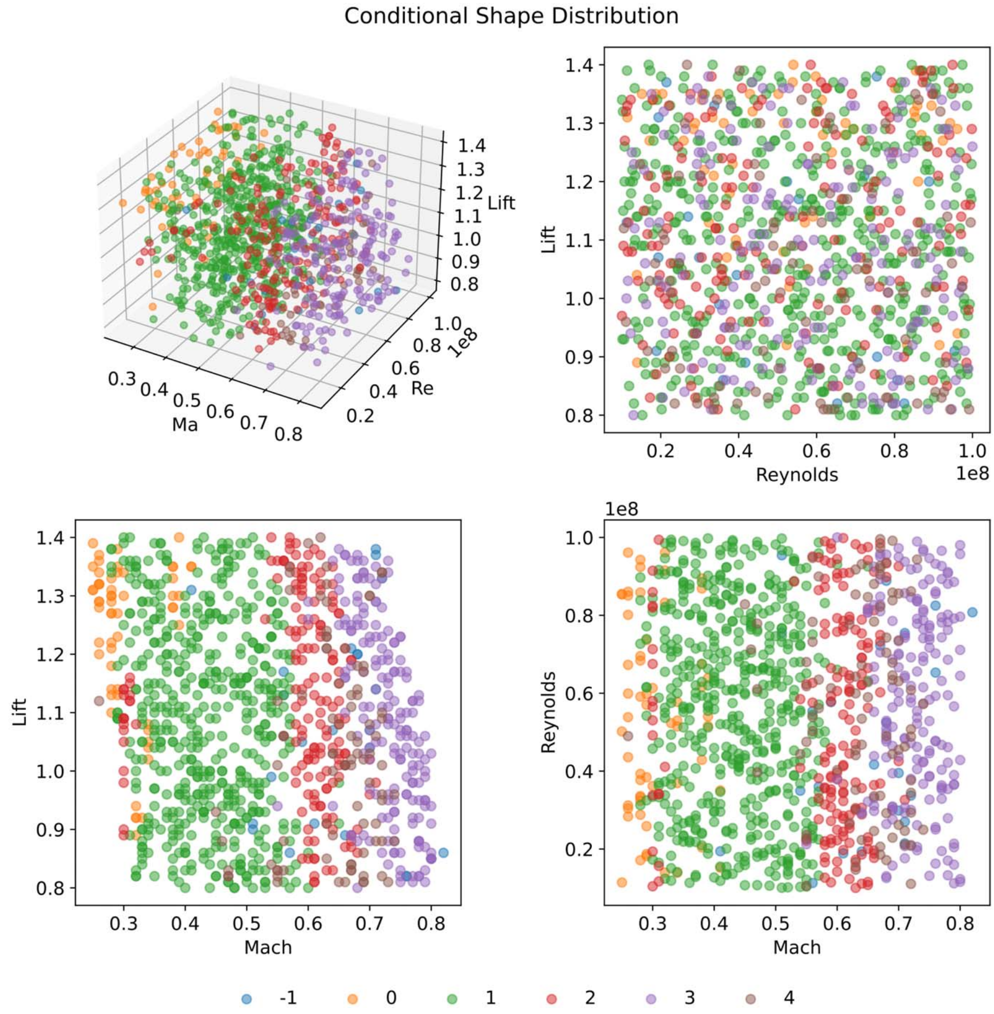

Fig. 8 Distribution of shape clusters with respect to Mach number, Reynolds number, and lift coefficient

optimal under a wide range of boundary conditions, compared with the others.

Despite seeming plausible, these two takeaways need to be taken with a grain of salt, as there are several caveats that may undermine their validity:

(1) The high variation along the AoA direction in Fig. 6 may not be attributed to the AoA alone, as some variations in the latent code may also have been projected along it under PCA. However, we think it is still safe to regard AoA as the dominant contributor to this direction's variation and ignore the airfoil shape's influence, as otherwise we should have already seen a similar linear pattern in Fig. 5(*a*), i.e., the PCA without AoA.

(2) The distance variation among the latent codes shown in Fig. 6 may not be directly comparable to variations in AoA. For instance, when comparing data A and data B, we may come up with a Mahalanobis distance function for A that scales up A's space arbitrarily large, such that the variation of data A measured by it is also arbitrarily large. It is then pointless to compare this variation with a normal Euclidean-based one of data B. In our case, the variations of shape and AoA are based on the Euclidean distances defined on the normalized latent space and normalized AoA space respectively. Because the latent set is isometric to the dataset, the variation of shape is also equivalently based on the geodesic distance on the shape manifold, up to the shape code's normalization factor.

(3) The previous caveat seems to not affect our second takeaway, because for that claim we only compare each cluster's shape variation (not including the AoA). The variation of shape, as mentioned above, is based on the geodesic distance between shapes. However, despite its awareness of the data manifold's geometry, whether or not the geodesic distance is reasonable for comparing shapes is still an open question. For instance, it might actually not be aligned with a human's "perceptual metric" [70], such that a large difference between airfoils in terms of the geodesic distance only corresponds to a small visual difference, and vice versa. In addition, sometimes a subtle visual change in shape may lead to a huge shift in the design's performance. Since the design's performance is what we care about ultimately, it might be more reasonable to compare shapes by measuring their difference in performance.

These concerns, together with our above takeaways, require future research. Nonetheless, it could be informative to investigate the third caveat tentatively by illustrating all airfoil shapes in each cluster. This allows us to see how varied they are visually and whether the degrees of visual variation agree with the Euclidean/geodesic based variations in Fig. 6. We can then assess, albeit qualitatively, if the geodesic distance agrees with our "perceptual metric."

To do this, for each cluster we superpose all its airfoils and plot them altogether on the left of Fig. 7. Visually, we see that clusters

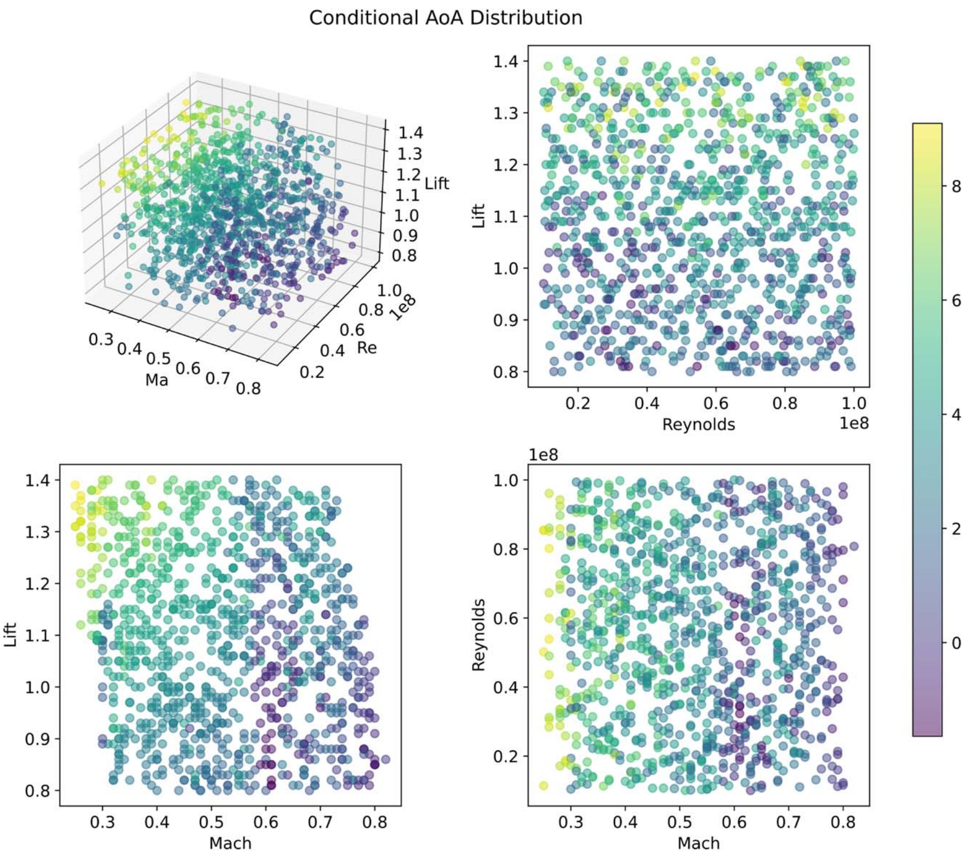

Fig. 9 Distribution of AoA with respect to Mach number, Reynolds number, and lift coefficient

#0 and #1 (Figs. 7(*a*) and 7(*c*)) have much lower variation in their airfoil shapes compared to clusters #2–#4 (Figs. 7(*e*), 7(*g*), and 7(*i*)). This agrees well with the variation difference shown in Fig. 6. This suggests the geodesic distance is at least a reasonable choice for evaluating the visual difference between shapes. On the right side of Fig. 7, we also plot the five initial airfoil designs—handpicked out of the eight restart candidates—that look the most similar to the airfoils in different clusters. We can see that for all clusters (maybe except #3), the optimized airfoils look almost identical to their corresponding initial designs, which testifies to both of our takeaways.

4.2.3 Relationships Between Airfoil Shape, Angle of Attack, and Boundary Condition. So far we have only analyzed the shape-AoA configurations of SU2 airfoils, without taking any boundary conditions into account. One potential hypothesis for the clustering of shapes in the latent space could be that each cluster corresponds to some identifiable change in the boundary conditions—i.e., that certain clusters arise naturally when optimizing a design within a range of conditions, and that the optimal cluster switches at some flow regime.

To investigate this hypothesis, we demonstrate the conditional distribution of airfoil shapes with respect to the three boundary condition parameters—Reynolds number, Mach number, and lift coefficient. Figure 8 scatter plots all the boundary conditions in the SU2 dataset, with each point marked according to its corresponding shape's cluster. We observe several features from this plot:

(1) There is a conspicuous laminated pattern along the Mach number dimension (bottom left and bottom right), which divides this dimension into five distinctive segments. Each segment is prominently occupied by the airfoil shapes in a single cluster. As the Mach number incrementally increases from ∼0.2 to ∼0.8, the optimal airfoil shape morphs from cluster to cluster following the order #0 → #1 → #2 → #4 → #3. Not only that, this morphing is also in general *monotonic* or *injective*, namely no airfoil cluster appears *dominantly* more than once in different Mach number segments.

(2) In contrast, the optimal shape is in general independent of Reynolds number and lift coefficient, as least within the SU2 dataset's cubic boundary condition regime. This is reflected in both how uniformly the shape clusters distribute in the Re–Lift subspace (top right), and how perpendicular the lamination boundaries are to the Ma dimension (bottom left and bottom right).

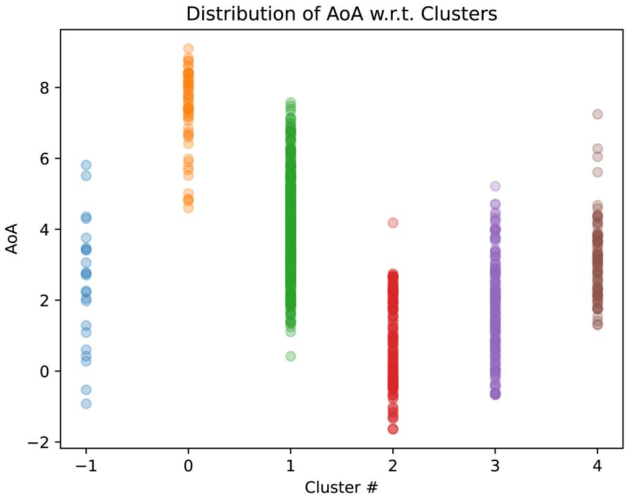

Fig. 10 Distribution of AoA with respect to clusters

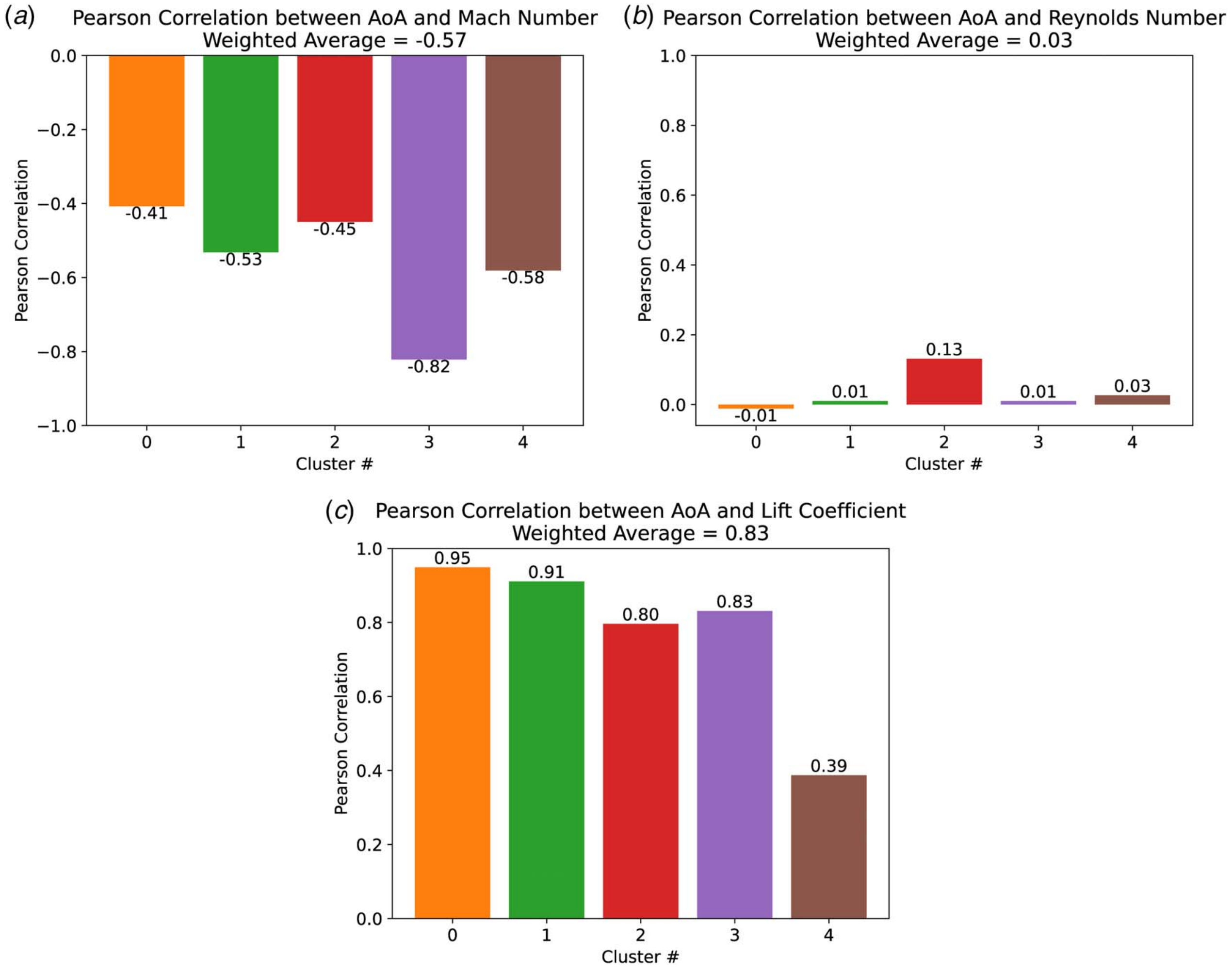

Fig. 11 Pearson correlations between AoA and boundary conditions in different clusters: (*a*) AoA—Ma, (*b*) AoA—Re, and (*c*) AoA—lift

(3) Again, not all airfoils are created equal. For example, cluster #1 dominates the wide flow velocity regime roughly between Mach 0.3 and 0.55, whereas cluster #4 concentrates around Ma 0.7, at the boundary between clusters #2 and #3.

Overall, in aerodynamic design optimization, the airfoil shape morphs primarily to adapt to its working velocity. This is consistent with existing design practice of customizing aerodynamic surfaces to different speed regimes (e.g., Boeing 737 versus Concorde). Not only that, as expected, an airfoil shape that is optimal within a given speed regime is not likely to be optimal within another. Moreover, different airfoil shapes also have different sensitivities to velocity, as some are adaptable to a wider range of speeds than others. It is intriguing that Fig. 8 found this known behavior in an unsupervised manner only through analyzing the learned latent space rather than being directly trained on the boundary conditions.

Figure 9 demonstrates whether this same pattern exists for changes in the AoA, with each point marked according to its AoA value. This graph shows that, in general, the optimal AoA is correlated not only with lift coefficient but also with Mach number. The latter, however, might be considerably affected by the shape lamination along the Ma dimension, as each cluster of shapes may have a distinctive optimal range of AoAs that couples with it, such that the AoA–Ma correlation mainly results from this cluster-wise coupling. Indeed, we can notice this coupling between shape cluster and AoA in Fig. 10.

To avoid the influence of this shape lamination, we instead study how AoA varies with respect to Ma, Re, and lift with the airfoil shape *fixed*. Specifically, we evaluate the Pearson correlation coefficients (PCC) between AoA and the three conditions in each cluster (recall that the shapes in each cluster look very similar, thus may be roughly regarded as fixed). The results are demonstrated in Fig. 11, which reveals AoA's moderate negative correlation with Mach number (PCC on average −0.57, weighted by cluster size), high positive correlation with lift coefficient (PCC on average 0.83, likewise), and negligible correlation with Reynolds number (PCC on average 0.03, likewise).

5 Limitations

The isometric embedding method has several limitations worth mentioning. First, we use the latent interpolation method described in Sec. 3.1 to tackle the potential dataset "pathologies" as discussed in Sec. 2.4. It should be noted that with this interpolation technique employed, we are preserving the geodesic distance on the *extended* data manifold instead of that on the original data manifold, which does not necessarily agree with each other. For instance, imagine we have a circle dataset $\mathbb{S}^1$ on a curved surface (2D manifold) embedded in a high-dimensional data space. If we train an isometric autoencoder of 2D latent space for this dataset (as this is the least dimensional Euclidean space for embedding $\mathbb{S}^1$), we are learning the geodesic distance on the curved surface rather than that on the circle, so the shortest path connecting two given points on the circle falls outside the circle. Nevertheless, this distance is supposedly still better than the Euclidean distance in the data space that takes no geometry into account.

In addition, despite the autoencoder's ability to do amortized inference on new unseen data samples, it may not reconstruct them well, which means the new samples and their latent set may not be homeomorphic to each other. This could undermine the validity of latent analyses relying on the dataset's topological properties. To guarantee that homeomorphicity, users need to update the autoencoder's parameters by retraining it over the new data samples to reduce the reconstruction error, which should not take long for a pre-trained model. It should be noted that a higher reconstruction loss over the new data samples does not necessarily mean the new data samples' latent set is not suitable for latent analysis. It could be the case that the encoder is still approximately an isometric topological embedding over the new data samples but the decoder's image is not aligned with the new samples.

6 Conclusions and Future Work

In this paper, we employed the isometric autoencoder to learn an isometric representation of the airfoil designs that preserve the geodesic distance. Then we performed distance-based analyses such as clustering in the isometric latent space to study different airfoil dataset's characteristics and complexities, while investigating the necessity and validity of preserving the geodesic distance in the latent space.

As to the preservation of the geodesic distance, we found in Sec 4.1.3 that without it the latent set produced by the autoencoder is at risk of comprising an arbitrary number of clusters, which may mislead engineers' interpretation of the design distribution. It is therefore necessary to impart isometricity to the autoencoder when analyzing data in its latent space. We also verified that the geodesic distance agrees well with our visual perception when it comes to detecting shape variation (at least for the airfoils, Sec. 4.2.2), hence it is reasonable to preserve it for shape analysis.

Through the lens of the isometric autoencoder and HDBSCAN clustering, we found that compared to the high-quality and diverse UIUC dataset, the SU2 dataset of optimized airfoils has far less variation in its airfoil shapes. This may be blamed on two culprits. First, when optimizing the airfoil's L/D efficiency, the SU2 adjoint optimizer prefers adjusting the AoA to morphing the airfoil shape, probably because this is more effective in increasing the L/D ratio in terms of the adjoint gradient (specifically, the L/D objective's gradient with respect to the AoA may have much larger norm than that with respect to the airfoil spline's control parameters). Second, when creating the SU2 dataset with the SU2 optimizer, for each boundary condition we only performed eight restarts from the eight candidate airfoils and kept only the best final design. This, paired with the first issue, leads to the lack of diversity in the SU2 dataset. A future way to improve the quality of the SU2 dataset would be to either introduce more candidate airfoils or replace the current adjoint-based optimizer.

Despite the SU2 dataset's diversity issue, analyzing it in the isometric latent space still provides many insights into the conditional distribution of optimized airfoils. It shows that the cruising speed is the primary factor in the design of airfoil shapes, and not all airfoil shapes are created equal, as some can work optimally in a broader range of speed. In addition, an airfoil shape that works optimally in one speed regime is not likely to do so in another—in other words, it is unlikely to find an airfoil that is universally optimal at every speed. Moreover, when the airfoil shape is determined, if we want to increase or maintain its lift coefficient while the speed goes down, the airfoil should pitch up. In conjunction, these two results suggest the condition distribution $p(\text{shape, AoA}|\text{Ma, Re, lift})$ that the inverse airfoil design models in Ref. [5] tried to capture might be factorized and simplified into $p(\text{AoA}|\text{lift, Ma, shape}) \cdot p(\text{shape}|\text{Ma})$. Although the insights on this specific domain are already known from past human efforts in airfoil design, what is unique is that the proposed Isometric Autoencoder uncovered these without explicitly being trained to do so, and that this technique can be applied to many other domains. This demonstrates the value of constructing isometry via Isometric AEs to new, more complex problems. We are investigating more complicated, high-dimensional design problems as one avenue of future work, and expecting more meticulous research on the sensitivity of the autoencoder and clustering method's hyperparameters.

Acknowledgment

We acknowledge the support from the National Science Foundation through award #1943699 as well as ARPA-E award DE-AR0001216.

Conflict of Interest

There are no conflicts of interest.

Data Availability Statement

The datasets generated and supporting the findings of this article are obtainable from the corresponding author upon reasonable request.

References

[1] Panchal, J. H., Fuge, M., Liu, Y., Missoum, S., and Tucker, C., 2019, "Machine Learning for Engineering Design," ASME J. Mech. Des., **141**(11), p. 110301.

[2] Missoum, S., and Tucker, C., 2019, "Topology Design With Conditional Generative Adversarial Networks," Volume 2A: 45th Design Automation Conference of International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, Anaheim, CA, Aug. 18–21.

[3] Yu, Y., Hur, T., Jung, J., and Jang, I. G., 2019, "Deep Learning for Determining a Near-Optimal Topological Design Without Any Iteration," Struct. Multidiscipl. Optim., **59**(3), pp. 787–799.

[4] Yilmaz, E., and German, B., 2020, "Conditional Generative Adversarial Network Framework for Airfoil Inverse Design," AIAA AVIATION 2020 FORUM, Virtual, June 15–19, American Institute of Aeronautics and Astronautics.

[5] Chen, Q., Wang, J., Pope, P., Chen, W., and Fuge, M., 2022, "Inverse Design of Two-Dimensional Airfoils Using Conditional Generative Models and Surrogate Log-Likelihoods," ASME J. Mech. Des., **144**(2), p. 021712.

[6] Wang, J., Chen, W. W., Da, D., Fuge, M., and Rai, R., 2022, "IH-GAN: A Conditional Generative Model for Implicit Surface-Based Inverse Design of Cellular Structures," Comput. Meth. Appl. Mech. Eng., **396**, p. 115060.

[7] Nobari, A. H., Chen, W., and Ahmed, F., 2021, "Range-Gan: Range-Constrained Generative Adversarial Network for Conditioned Design Synthesis.

[8] Yang, Z., Li, X., Catherine Brinson, L., Choudhary, A. N., Chen, W., and Agrawal, A., 2018, "Microstructural Materials Design Via Deep Adversarial Learning Methodology," ASME J. Mech. Des., **140**(11), p. 111416.

[9] Mosser, L., Dubrule, O., and Blunt, M. J., 2017, "Reconstruction of Three-Dimensional Porous Media Using Generative Adversarial Neural Networks," Phys. Rev. E, **96**, p. 043309.

[10] Xue, T., Wallin, T. J., Menguc, Y., Adriaenssens, S., and Chiaramonte, M., 2020, "Machine Learning Generative Models for Automatic Design of Multi-Material 3D Printed Composite Solids," Ext. Mech. Lett., **41**, p. 100992.

[11] Chen, W., Chiu, K., and Fuge, M., 2019, "Aerodynamic Design Optimization and Shape Exploration Using Generative Adversarial Networks," AIAA Scitech 2019 Forum, San Diego, CA, Jan. 7–11, p. 2351.

[12] Chen, W., and Ahmed, F., 2021, "PaDGAN: Learning to Generate High-Quality Novel Designs," ASME J. Mech. Des., **143**(3), p. 031703.

[13] Liang, L., Liu, M., Martin, C., and Sun, W., 2018, "A Deep Learning Approach to Estimate Stress Distribution: A Fast and Accurate Surrogate of Finite-Element Analysis," J. R. Soc. Interface., **15**(138), p. 20170844.

[14] Jiang, H., Nie, Z., Yeo, R., Farimani, A. B., and Kara, L. B., 2021, "StressGAN: A Generative Deep Learning Model for Two-Dimensional Stress Distribution Prediction," ASME J. Appl. Mech., **88**(5), p. 051005.

[15] Nie, Z., Jiang, H., and Kara, L. B., 2019, "Stress Field Prediction in Cantilevered Structures Using Convolutional Neural Networks," ASME J. Comput. Inf. Sci. Eng., **20**(1), p. 011002.

[16] Kochkov, D., Smith, J. A., Alieva, A., Wang, Q., Brenner, M. P., and Hoyer, S., 2021, "Machine Learning–Accelerated Computational Fluid Dynamics," Proc. Natl. Acad. Sci. U. S. A., **118**(21), p. e2101784118.

[17] Kim, B., Azevedo, V. C., Thuerey, N., Kim, T., Gross, M., and Solenthaler, B., 2019, "Deep Fluids: A Generative Network for Parameterized Fluid Simulations," Comput. Graph. Forum, **38**(2), pp. 59–70.
[18] Bellman, R., 2010, *Dynamic Programming*, Princeton University Press, Princeton, NJ.
[19] Belkin, M., and Niyogi, P., 2001, "Laplacian Eigenmaps and Spectral Techniques for Embedding and Clustering," *Advances in Neural Information Processing Systems*, Vol. 14, T. Dietterich, S. Becker, and Z. Ghahramani, eds., MIT Press, Cambridge, MA.
[20] Van der Maaten, L., and Hinton, G., 2008, "Visualizing Data Using t-SNE," J. Mach. Learn. Res., **9**(11), pp. 2579–2605.
[21] McInnes, L., Healy, J., and Melville, J., 2018, "UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction," arXiv.
[22] Chen, W., Fuge, M., and Chazan, J., 2017, "Design Manifolds Capture the Intrinsic Complexity and Dimension of Design Spaces," ASME J. Mech. Des., **139**(5), p. 051102.
[23] Chen, N., Klushyn, A., Ferroni, F., Bayer, J., and Van Der Smagt, P., 2020, "Learning Flat Latent Manifolds With Vaes," Proceedings of the 37th International Conference on Machine Learning, Virtual, July 13–18.
[24] Gropp, A., Atzmon, M., and Lipman, Y., 2020, "Isometric Autoencoders," arXiv.
[25] Yonghyeon, L., Yoon, S., Son, M., and Park, F. C., 2022, "Regularized Autoencoders for Isometric Representation Learning," International Conference on Learning Representations, Virtual, Apr. 22–25.
[26] Ester, M., Kriegel, H.-P., Sander, J., and Xu, X., 1996, "A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases With Noise," KDD, Vol. 96, 226–231.
[27] Campello, R. J., Kröger, P., Sander, J., and Zimek, A., 2020, "Density-Based Clustering," Wiley Interdiscip. Rev.: Data Min. Knowl. Discov., **10**(2), p. e1343.
[28] SIGKDD, A., 2014, "Sigkdd Test of Time Award."
[29] Schubert, E., Sander, J., Ester, M., Kriegel, H. P., and Xu, X., 2017, "Dbscan Revisited, Revisited: Why and How You Should (Still) Use DBSCAN," ACM Trans. Database Syst. (TODS), **42**(3), pp. 1–21.
[30] Gan, J., and Tao, Y., 2015, "Dbscan Revisited: Mis-Claim, Un-Fixability, and Approximation," Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data, Melbourne, VIC, Australia, May 31–June 4, pp. 519–530.
[31] Ankerst, M., Breunig, M. M., Kriegel, H.-P., and Sander, J., 1999, "Optics: Ordering Points to Identify the Clustering Structure," ACM Sigmod Rec., **28**(2), pp. 49–60.
[32] Hinneburg, A., and Keim, D. A., 1998, *An Efficient Approach to Clustering in Large Multimedia Databases With Noise*, Vol. 98, Bibliothek der Universität Konstanz, Konstanz, Germany.
[33] McInnes, L., Healy, J., and Astels, S., 2017, "HDBSCAN: Hierarchical Density Based Clustering," J. Open Source Softw., **2**(11), p. 205.
[34] Johnson, S. C., 1967, "Hierarchical Clustering Schemes," Psychometrika, **32**(3), pp. 241–254.
[35] MacQueen, J., 1967, "Classification and Analysis of Multivariate Observations," Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability Held at the Statistical Laboratory University of California, Los Angeles, CA, pp. 281–297.
[36] Bishop, C. M., and Nasrabadi, N. M., 2006, *Pattern Recognition and Machine Learning*, Vol. 4, Springer, New York.
[37] Andoni, A., 2009, "Nearest Neighbor Search: the Old, the New, and the Impossible," Ph.D. Thesis, Massachusetts Institute of Technology, Cambridge, MA,
[38] Kulesza, A., and Taskar, B., 2012, "Determinantal Point Processes for Machine Learning," Found. Trends. Mach. Learn., **5**(2–3), pp. 123–286.
[39] Ahmed, F., and Fuge, M., 2018, "Ranking Ideas for Diversity and Quality," ASME J. Mech. Des., **140**(1), p. 011101.
[40] Beyer, K., Goldstein, J., Ramakrishnan, R., and Shaft, U., 1999, "When is 'Nearest Neighbor' Meaningful?," Database Theory–ICDT'99: 7th International Conference, Proceedings 7, Jerusalem, Israel, Jan. 10–12, Springer, pp. 217–235.
[41] Aggarwal, C. C., Hinneburg, A., and Keim, D. A., 2001, "On the Surprising Behavior of Distance Metrics in High Dimensional Space," Database Theory–ICDT 2001: 8th International Conference, Proceedings 8, London, UK, Jan. 4–6, Springer, pp. 420–434.
[42] Weinberger, K. Q., and Saul, L. K., 2009, "Distance Metric Learning for Large Margin Nearest Neighbor Classification," J. Mach. Learn. Res., **10**(2), pp. 207–244.
[43] Fefferman, C., Mitter, S., and Narayanan, H., 2016, "Testing the Manifold Hypothesis," J. Am. Math. Soc., **29**(4), pp. 983–1049.
[44] Roth, P. M., Hirzer, M., Köstinger, M., Beleznai, C., and Bischof, H., 2014, "Mahalanobis Distance Learning for Person Re-Identification," *Person Re-Identification*, Springer, London, UK, pp. 247–267.
[45] Lee, K., Lee, K., Lee, H., and Shin, J., 2018, "A Simple Unified Framework for Detecting Out-of-Distribution Samples and Adversarial Attacks," Adv. Neural Inf. Process. Syst., **31**, pp. 7167–7177.
[46] Bronstein, M. M., Bruna, J., Cohen, T., and Veličković, P., 2021, "Geometric Deep Learning: Grids, Groups, Graphs, Geodesics, and Gauges," arXiv.
[47] Lee, J. M., 2006, *Riemannian Manifolds: An Introduction to Curvature*, Vol. 176, Springer Science & Business Media, New York, NY.
[48] Fletcher, P. T., Lu, C., Pizer, S. M., and Joshi, S., 2004, "Principal Geodesic Analysis for the Study of Nonlinear Statistics of Shape," IEEE Trans. Med. Imag., **23**(8), pp. 995–1005.
[49] Nilsson, J., Fioretos, T., Höglund, M., and Fontes, M., 2004, "Approximate Geodesic Distances Reveal Biologically Relevant Structures in Microarray Data," Bioinformatics, **20**(6), pp. 874–880.
[50] Srivastava, A., Klassen, E., Joshi, S. H., and Jermyn, I. H., 2010, "Shape Analysis of Elastic Curves in Euclidean Spaces," IEEE Trans. Pattern Anal. Mach. Intell., **33**(7), pp. 1415–1428.
[51] Tosi, A., Hauberg, S., Vellido, A., and Lawrence, N. D., 2014, "Metrics for Probabilistic Geometries," Proceedings of the Thirtieth Conference on Uncertainty in Artificial Intelligence, Quebec City, Quebec, Canada, July 23–27.
[52] Chen, N., Klushyn, A., Kurle, R., Jiang, X., Bayer, J., and Smagt, P., 2018, "Metrics for Deep Generative Models," International Conference on Artificial Intelligence and Statistics, Playa Blanca, Lanzarote, Canary Islands, Apr. 9–11, PMLR, pp. 1540–1550.
[53] Linderman, G. C., and Steinerberger, S., 2019, "Clustering With t-SNE, Provably," SIAM J. Math. Data Sci., **1**(2), pp. 313–332.
[54] Tenenbaum, J. B., Silva, V. d., and Langford, J. C., 2000, "A Global Geometric Framework for Nonlinear Dimensionality Reduction," Science, **290**(5500), pp. 2319–2323.
[55] Roweis, S. T., and Saul, L. K., 2000, "Nonlinear Dimensionality Reduction by Locally Linear Embedding," Science, **290**(5500), pp. 2323–2326.
[56] Goodfellow, I., Bengio, Y., and Courville, A., 2016, *Deep Learning*, MIT Press, Cambridge, MA.
[57] Coifman, R. R., and Lafon, S., 2006, "Diffusion Maps," Appl. Comput. Harmon. Anal., **21**(1), pp. 5–30.
[58] Lee, J., 2010, *Introduction to Topological Manifolds*, Vol. 202, Springer Science & Business Media, New York.
[59] Asinomàs. Composition Equals Identity Function. Mathematics Stack Exchange.
[60] Dinh, L., Krueger, D., and Bengio, Y., 2015, "Nice: Non-Linear Independent Components Estimation," 3rd International Conference on Learning Representations, San Diego, CA, May 7–9.
[61] Dinh, L., Sohl-Dickstein, J., and Bengio, S., 2016, "Density Estimation Using Real NVP," 5th International Conference on Learning Representations, Toulon, France, Apr. 24–26.
[62] Kingma, D. P., and Dhariwal, P., 2018, "Glow: Generative Flow With Invertible 1 × 1 Convolutions," Adv. Neural Inf. Process. Syst., **31**, pp. 10215–10224.
[63] Zhang, H., Cisse, M., Dauphin, Y. N., and Lopez-Paz, D., 2018, "Mixup: Beyond Empirical Risk Minimization," 6th International Conference on Learning Representations, Vancouver, BC, Canada, Apr. 30– May 3.
[64] Zhu, J.-Y., Park, T., Isola, P., and Efros, A. A., 2017, "Unpaired Image-to-Image Translation Using Cycle-Consistent Adversarial Networks," Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, Oct. 22–29.
[65] Lee, J. M., and Lee, J. M., 2012, *Smooth Manifolds*, Springer, New York, NY.
[66] Levina, E., and Bickel, P., 2004, "Maximum Likelihood Estimation of Intrinsic Dimension," Adv. Neural Inf. Process. Syst., **17**, pp. 777–784.
[67] MacKay, D. J., and Ghahramani, Z., 2005, "Comments on 'Maximum Likelihood Estimation of Intrinsic Dimension" by E. *Levina and P. Bickel*.
[68] Chen, Q., Pope, P., and Fuge, M., 2022, "Learning Airfoil Manifolds With Optimal Transport," AIAA SCITECH 2022 Forum, San Diego, CA, Jan. 3–7, p. 2352.
[69] Feydy, J., Séjourné, T., Vialard, F.-X., Amari, S.-i., Trouvé, A., and Peyré, G., 2019, "Interpolating Between Optimal Transport and Mmd Using Sinkhorn Divergences," The 22nd International Conference on Artificial Intelligence and Statistics, Naha, Okinawa, Japan, Apr. 16 –18, PMLR, pp. 2681–2690.
[70] Zhang, R., Isola, P., Efros, A. A., Shechtman, E., and Wang, O., 2018, "The Unreasonable Effectiveness of Deep Features as a Perceptual Metric," Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, June 18–22, pp. 586–595.