AIAA SciTech Forum
January 3-7, 2022, San Diego, CA & Virtual
AIAA SCITECH 2022 Forum

10.2514/6.2022-2352

Learning Airfoil Manifolds with Optimal Transport

Qiuyi Chen* Phillip Pope† Mark Fuge‡
University of Maryland, College Park, Maryland, 20720

The manifold hypothesis forms a pillar of many modern machine learning techniques. Within the context of design, it proposes that valid designs reside on low dimensional manifolds in the high dimensional design spaces. Our previous research—BézierGAN—suggests learning the low dimensional parameterization of valid airfoil designs can indeed accelerate aerodynamic optimizations. However, it incurs problems such as misalignment, long training time, and the trouble of finding the latent dimensionality. In this work we present the optimal-transport-based sibling of BézierGAN that surpass its predecessor in terms of both manifold approximating precision and learning speed, and provide methodology that helps determine the intrinsic dimension of the design manifold beforehand.

I. Introduction

Aerodynamic shape optimization is essential for designing airfoils. It begins with shape parameterization [1], whereby the airfoil contour is delineated by certain spline governed by a finite number (n) of design parameters. In order to precisely represent any airfoil, n is usually set very large, so much so that most configurations in the n-D design space correspond to disordered curves. In consequence, this oversized design space impedes the optimizations on it due to the curse of dimensionality [2], and hence decelerate the development of airfoils.

To tackle this issue, researchers have been looking into dimension reduction methods [3–12]. The motivation stems from the manifold hypothesis [13, 14], namely to retrieve from this n-D design space a submanifold of much lower dimensionality m, such that all or most configurations in this new m-D space corresponds to valid airfoils, thereby the optimization on it is easier. The recent work of BézierGAN [11] in particular leverages the universal approximation ability of neural networks to capture the nonlinear airfoil manifold from UIUC airfoil database.

Yet one critical concern is how well the approximate manifold aligns with the exact one. It is possible that they do not overlap well, such that in the subsequent shape optimization, those airfoils the approximate manifold misses will never be explored and we can only get suboptimal results. BézierGAN's minimax training process is widely reported to induce mode collapse [15], which suggests misalignment. Apart from this problem, BézierGAN also incurs long training time ascribed to the training phase for its discriminator, and is bothered by tuning the latent dimensionality m.

In this work, we aim to alleviate the misalignment by replacing the minimax game in BézierGAN with an objective based on the recent development in computational optimal transport (OT) [16], such that the airfoil distribution can be better captured (through the lens of our metrics) while accelerating the training process. In addition, we substitute its variational mutual information maximization [17] with the direct maximization of a new OT dependency measure that helps achieve the same "interpretable" latent space but reduce the training time. Moreover, we study the practicability of intrinsic dimension estimation techniques [18] that may help determining the size of latent dimension beforehand, thus facilitate the design of the neural network generator.

Specifically, this conference paper includes the following contributions:

1) We review Optimal Transport and specifically the Sinkhorn Divergence as an alternative means of optimizing Generative Models, and detail its advantages compared to traditional GAN approaches, including training speed and robustness to mode collapse.
2) We define a Sinkhorn Dependency Measure (SDM) that can substitute for the KL divergence based Mutual Information within a generative model. We experimentally demonstrate how this produces more interpretable latent space representations of airfoils than traditional InfoGAN architectures.
3) We propose a method for determining the intrinsic dimension of low-dimensional manifolds via Maximum Likelihood Estimate, and demonstrate that this matches the intrinsic dimension found through brute force hyper-parameter tuning, but at a fraction of the computational and experimental cost.

*PhD Student, Department of Mechanical Engineering
†PhD Student, Department of Computer Science
‡Associate Professor, Department of Mechanical Engineering

4) We illustrate and analyze the effect of latent dimension, SDM maximization and mode collapse in intuitive low dimensional settings to provide insight into their effect on high dimensional manifold learning.

II. Background

A. Manifold Learning

Although the term *manifold learning* has yet to assume a unanimous definition in the machine learning community, to obviate ambiguity in this optimization related work, we narrow its definition down to the process of *learning a parameterization map $g : \mathcal{Z} \to X$ from a regular subset $\mathcal{Z}$ in a latent Euclidean space Z to the high dimensional data space X given some data points sampled from the low dimensional data manifold $\mathcal{X} \subset X$, such that*

- *At best $\mathcal{X} \subseteq \operatorname{Im} g$, and the complement $\operatorname{Im} g \setminus \mathcal{X}$ is as close to $\emptyset$ as possible (i.e., $\operatorname{Im} g$ tightly covers $\mathcal{X}$),*
- *Or practically at least $\sup_{x\in\mathcal{X}} \inf_{x'\in \operatorname{Im} g} d(x, x') \leq \epsilon$ for given metric d and $\epsilon > 0$, while $\operatorname{Im} g$ is as small as possible regarding certain measure μ (i.e., $\operatorname{Im} g$ is sufficiently close to $\mathcal{X}$).*

Here we call $\mathcal{Z}$ *regular* if it is a single or the union of a finite number of simply connected sets of non-zero measure, such that we can move inside it and sample points from it with ease. This definition is founded on the bold but widely-adopted *manifold hypothesis* [13] that all valid data points constitute a low m-dimensional manifold—a set locally homeomorphic to an m-D Euclidean space at every point—that is embedded in the high n-dimensional data space. It entails the considerable opportunity for us to explore the data manifold more efficiently by exploring the regular subset in the usually *low* dimensional latent space.

So how is this manifold learning associated with airfoil optimization, or any shape optimization? The task of optimization is to find the design optima $x^\star$ in the space X of shape parameters given certain objective function. Usually in order to obtain a shape parameterization of sufficient expressivity, X is very high dimensional such that most $x \in X$ only correspond to invalid eccentric shapes. It is then tempting to assume that all valid designs reside on an unknown low dimensional manifold $\mathcal{X} \subset X$, and to find $x^\star \in \mathcal{X}$ we only need to explore on or close to $\mathcal{X}$ by optimizing over $\mathcal{Z} \subseteq Z$ where $\dim Z \ll \dim X$, provided we can find the corresponding parameterization $g : \mathcal{Z} \to X$ whose image *learned* $\mathcal{X}$ well in terms of the aforementioned definition. Because of this, we want $\operatorname{Im} g \setminus \mathcal{X}$ or equivalently $\operatorname{Im} g$ to be as small as possible to avert getting z mapped to x too far away from $\mathcal{X}$, and $\mathcal{Z}$ to be regular to reduce that odds of $z + \Delta z$ stepping out of $\mathcal{Z}$ for a $z \in \operatorname{int} \mathcal{Z}$ and a small enough Δz. The latter comes in very handy when doing gradient based optimization. It would be even better if $\mathcal{Z}$ is convex as it enables linear interpolation.

B. Generative Models for Manifold Learning

Many existing algorithms fall within the scope of this manifold learning definition. For instance, PCA [19] is basically a manifold learning method with g being a linear map and $\mathcal{Z}$ being the Euclidean principal component space, so its image is a hyperplane not necessarily covering, but intersecting the generally nonlinear data manifold, which implies $\mathcal{X} \nsubseteq \operatorname{Im} g$ and explains its poor performance on highly nonlinear data. Autoencoder [20] learns this g with an encoder $e : \mathcal{X} \to Z$ by minimizing $\mathbb{E}_{x\sim\mathcal{X}}\|g \circ e(x) - x\|$, *i.e.*, forcing $g \circ e(x)$ to become an *identity* map on $\mathcal{X}$, which is only possible when e is *injective* and g is *surjective*, and this indicates $\mathcal{X} \subseteq \operatorname{Im} g$. However, because e is not necessarily surjective and $\mathcal{Z} = \operatorname{Im} e$ could be of zero measure in the latent space Z thus irregular, if we just randomly sample z from Z and feed it to g, the result $g(z)$ is very likely to fall in $\operatorname{Im} g \setminus \mathcal{X}$, so the decoder g is not of practical use in general.

Recently developed unsupervised learning algorithms such as GANs [21–25], VAEs [26, 27] and flow-based models [28–30] also count as manifold learning algorithms from this perspective, as they are all used to train a generator $g : \mathcal{Z} \to X$, where $\mathcal{Z}$ is often a hypersphere supporting Gaussian distribution or a hypercube supporting uniform distribution or a mix of these two. However, we can ignore flow-based model here because its generator has image of non-zero measure in the data space X, which is to large for learning low dimensional $\mathcal{X}$. The generators in GANs and VAEs after training are supposed to generate samples with distribution p_g similar to the dataset distribution p_r via transforming the latent distribution p_Z over $\mathcal{Z}$. This is done in two simultaneous steps. First, the generator g must map almost every point on $\mathcal{Z}$ to a structure in X very close to or covering $\mathcal{X}$. Second, g makes this structure dilate or shrink itself locally via adjusting $|[\nabla g]^T[\nabla g]|^{\frac{1}{2}}$ to have the points it generates more condensed in higher density regions and more sporadic in lower density regions [31]. The first step is exactly what we want for the parameterization of the manifold $\mathcal{X}$, whereas the second step is unnecessary if not detrimental to our optimization application as all designs on $\mathcal{X}$ should be equally valid. However, these two steps are seamlessly interwoven with each other in GANs' and VAEs' probabilistic training method—namely minimizing some statistical distance or maximizing sample log likelihood lower

bound—and thus pitifully we cannot isolate the useful effect of the first step. In this work we will focus on a optimal transport variant of the vanilla GAN [21] to train a desired parameterization g. Vanilla GAN trains the generator g by performing the vanilla minimax game $\min_g \max_d \ \mathbb{E}_{\mathbf{x}\sim p_r(\mathbf{x})}[\log d(\mathbf{x})] + \mathbb{E}_{\mathbf{z}\sim p_Z(\mathbf{z})}[\log(1 - d(g(\mathbf{z})))]$, where d is a discriminator differentiating between real and fake samples drawn from p_r and p_g.

III. Optimal Transport and Sinkhorn Divergence

Training GAN with the vanilla minimax game, as in BézierGAN [11], is approximately minimizing the JS divergence $\mathrm{JSD}(p_r \mid p_g)$ between the real data distribution p_r and the approximate distribution p_g represented by the generator g [21]. Because of manifold misalignment, this minimax game is prone to induce convergence and stability issues [32]. It can also lead to mode collapse as its training objective is biased [15]. To circumvent these, researcher have turned to optimal transport statistical distances in place of JS divergence [24, 25]. The recent development of entropy regularization further facilitates their implementation [33–35].

The entropy regularized optimal transport distance between two distributions p_r and p_g is defined by [36]:

$$\mathrm{OT}_\lambda(p_r \mid p_g) \coloneqq \min_{\mathbb{P}_{X,\hat{X}} \in \Pi(p_r, p_g)} \mathbb{E}_{\mathbf{x},\hat{\mathbf{x}}\sim\mathbb{P}_{X,\hat{X}}}[l(\mathbf{x}, \hat{\mathbf{x}})] + \lambda \mathrm{KL}(\mathbb{P}_{X,\hat{X}} \mid p_r \times p_g) \tag{1}$$

where $\Pi(p_r, p_g)$ is the set of joint distributions $\mathbb{P}_{X,\hat{X}}$ whose marginal distributions equal p_r and p_g, l is a cost function usually symmetric positive, and $\lambda \geq 0$ is a coefficient for the degree of regularization. Though hard to evaluate, thanks to strong duality, OT_λ can be calculated instead via its dual

$$\mathrm{OT}_\lambda(p_r \mid p_g) = \max_{\alpha,\beta} \mathbb{E}_{\mathbf{x}\sim p_r}[\alpha(\mathbf{x})] + \mathbb{E}_{\hat{\mathbf{x}}\sim p_g}[\beta(\hat{\mathbf{x}})] - \lambda \mathbb{E}_{\mathbf{x},\,\hat{\mathbf{x}}\sim p_r \times p_g}\left[\exp\frac{\alpha(\mathbf{x}) + \beta(\hat{\mathbf{x}}) - l(\mathbf{x}, \hat{\mathbf{x}})}{\lambda} - 1\right] \tag{2}$$

where α and β are the Lagrange multipliers and their optimum can be rapidly found with Sinkhorn algorithm [36]. The cost function l is usually set to distance functions, such as L_1 and L_2 distance, and λ needs to be tuned on a trial and error basis, although heuristically we can just set it to a value between 0 and 1. In order to eliminate the intrinsic bias of OT_λ (*i.e.*, $\mathrm{OT}_\lambda(p \mid p) \neq 0$ for $\lambda > 0$) that leads to mode collapse, *Sinkhorn divergence* [36] is introduced, defined by

$$\mathrm{S}_\lambda(p_r \mid p_g) \coloneqq \mathrm{OT}_\lambda(p_r \mid p_g) - \frac{1}{2}\mathrm{OT}_\lambda(p_r \mid p_r) - \frac{1}{2}\mathrm{OT}_\lambda(p_g \mid p_g) \tag{3}$$

We can then train the generator by minimizing $\mathrm{S}_\lambda(p_r \mid p_g)$ to make p_r and p_g identical.

The convergence difference between the vanilla minimax game and the Sinkhorn divergence will be inspected with MMD [37] equiped with Gaussain kernel, which can measure the discrepancy between distributions without knowing the probability density function. We will also examine their robustness against mode collapse and discuss its implication for manifold learning via low dimensional toy problems in §VI.C, given that this issue is not as easy to reveal and examine in high dimensional cases.

IV. Mutual Information and Sinkhorn Dependency Measure

The same manifold can be expressed by different parameterizations. We prefer a parameterization that is "interpretable" or "disentangled", which, we propose, actually means the mapping from the latent subset to the high dimensional objects on the manifold is as *injective* as possible such that the shape changes monotonically along any direction in the latent space and all the major shape variations are captured. In other words, g had better be a homeomorphism. This reduces the number of optima and makes the optimization over the manifold less complicated.

We may decompose the latent Euclidean space Z as a product of two arbitrary Euclidean subspaces, namely $Z = C \times N$, where C can be called *latent code space* and N *latent noise space*. To make at least the latent code c in the subspace C "interpretable", mutual information maximization is commonly practiced in representation learning tasks, such as InfoGAN [17] and its descendant BézierGAN [11], where a lower bound of the mutual information $I(c; g(c, n)) = \mathrm{KL}(p(x \mid c)p(c) \mid p(x)p(c))$ is maximized since the exact evaluation is infeasible. Yet the variational gap is usually not tight thereby detrimental to the final performance [38], not to mention that the evaluation of the lower bound involves the training of a discriminator and consumes a lot of time. To avoid these problems, we replace the KL divergence with the aforementioned Sinkhorn divergence to define the *Sinkhorn dependency measure* (SDM) as follows, which can be evaluated directly with efficiency:

$$I_S(c; g(c, n)) \coloneqq \mathrm{S}_\lambda(p(x \mid c)p(c) \mid p(x)p(c)) \tag{4}$$

A BézierGAN equipped with these two new training mechanisms is dubbed *Entropic BézierGAN*, which is trained by minimizing $\mathrm{S}_\lambda(p_r \mid p_g) - \gamma \mathrm{S}_\lambda(p_{x,c} \mid p_x p_c)$ where γ is a weight coefficient. Though not yet guaranteed in theory to have the same effect, we will show experimentally on the airfoil dataset that maximizing this new dependency measure leads to the same interpretable airfoil representation in much shorter time, via the visual inspection of slices of the latent space and the *Latent Space Consistency* metric [39], as shown in Figure 7 and Table 1 of the experiment section.

V. Intrinsic Dimension Estimation

One critical hyperparameter that needs to be determined for manifold learning is the dimensionality of the latent space, or, in other words, the dimensionality of the data manifold. Although it can *sometimes* be progressively assessed by trial and error after repetitively training generators of different architectures, such great expense can be spared if there is an efficient way to determine it beforehand.

A priori the geometry of airfoil manifolds is unknown and may be complex. Several challenges arise when computing intrinsic dimensionality estimates. First is deciding the appropriate length scale of the problem, encoded by the hyperparameter k. Second is the number of samples required to estimate the manifold. One may not have enough samples to fully characterize the underlying manifold. For instance, some works in the recent literature suggest it is difficult to estimate image manifolds of intrinsic dimension greater than 32 with less than $1M$ samples [18]. Lastly, we note estimators like MLE rely on the assumptions which may not hold in practice, such as the data density being locally uniform and sufficiently smooth, and that the data is IID. The truth of these assumptions is unknown for airfoil manifolds.

With these caveats in mind, we proceed to use the *Maximum Likelihood Estimation* (MLE) of [40] and with the bias correction suggested by [41] as our intrinsic dimension estimator for the UIUC airfoil dataset. We now briefly describe the MLE estimator. For complete details see [40]. Like most dimensionality estimators, MLE is based on k nearest neighbor statistics. Given a point x in the data domain and assuming that the local density is constant, and that the number of points within a small radius of a point follows a Poisson distribution, maximization of the likelihood equations yields the following estimate

$$\hat{m}_k(x) = \left[\frac{1}{k-1} \sum_{j=1}^{k-1} \log \frac{T_k(x)}{T_j(x)} \right]^{-1} \tag{5}$$

where $T_j(x)$ is the Euclidean distance from x to its j^{th} nearest neighbor. This obtains an local estimate for each data point, which must be combined in some way. In their original paper [40] propose to average these local estimates to obtain a global estimate $\bar{m}_k = \frac{1}{n} \sum_{i=1}^{n} \hat{m}_k(x_i)$. However, [41] suggest a debiasing correction with lower variance based on averaging of *inverses* instead

$$\bar{m}_k = \left[\frac{1}{n} \sum_{i=1}^{n} \hat{m}_k(x_i)^{-1} \right]^{-1} = \left[\frac{1}{n(k-1)} \sum_{i=1}^{n} \sum_{j=1}^{k-1} \log \frac{T_k(x_i)}{T_j(x_i)} \right]^{-1} \tag{6}$$

We use Equation 6 as our primary estimator. To carry out the MLE estimation, we follow the procedure of [18], where convergence is evaluated by running MLE estimation for a range of random samples sizes up to the dataset size. This convergence analysis helps to evaluate if enough data is available for the estimation. In addition, a variety of parameters k, the number of nearest neighbors used in the calculation, are used to evaluate the length-scale of the problem.

To further validate our estimates, we also try other popular intrinsic dimensionality estimators in the literature, namely the TwoNN and GeoMLE methods from [42] and [43]. These methods are variants on the MLE method but slightly different properties. For instance, the GeoMLE method attempts to relax the local uniformity assumption of MLE. For further details of these methods we refer the reader to the original papers.

VI. Experiment: Low Dimensional Manifold Learning Problems

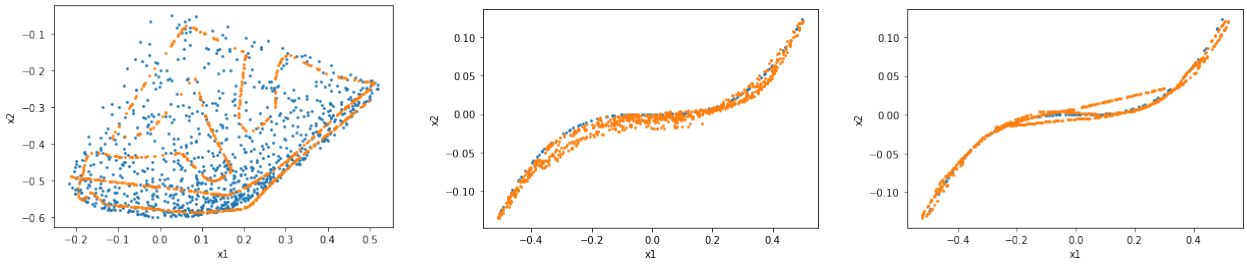
In this section we first demonstrate the significant effect of the generator's latent dimension on manifold learning in 2D setting and discuss its implications for its counterpart in higher dimensions. Thereafter we reveal the effect of mutual information maximization on manifold learning, and put forward the proposition that the "disentangled representation"

is essentially the embodiment of the regularized generator's *injectivity*. At last we examine the effect of mode collapse and discuss the importance of avoiding it for manifold learning.

A. Effect of Latent Dimension on Manifold Learning

Suppose we have an ANN generator $g : \mathcal{Z} \to X$ that is a smooth map (or almost everywhere smooth) from a regular subset $\mathcal{Z}$ in the latent space Z to the data space X where the data manifold resides. For $\dim X \geq \dim Z$, we know that the Jacobian ∇g, if exists, cannot have rank more than the latent dimension $\dim Z$. If the Jacobian is injective at $z \in \mathcal{Z}$, *i.e.*, $\text{rank}\nabla g(z) = \dim Z$, then according to the *Inverse Function Theorem* and the *Local Immersion Theorem* [44], the image of g is locally a manifold of dimension $\dim Z$ in X. This conclusion indicates that we cannot represent a manifold $\mathcal{X} \subseteq X$ with a generator whose latent dimension is lower than $\dim \mathcal{X}$. Intuitively speaking, we cannot fit a surface with a curve.

We use three examples to concretely demonstrate this. First, we construct a 2D manifold in $\mathbb{R}^2$ with a nonlinear diffeomorphism $\phi : [0, 1]^2 \to \mathbb{R}^2$ which is an MLP with randomly sampled non-singular weight matrices and tanh activations. Specifically, we uniformly sample 1000 points from the square $[0, 1]^2$ and transform them via ϕ to obtain the new samples on the target 2D manifold. We then train an EGAN on this dataset, whose generator is of latent dimension 1 and the latent distribution is the uniform distribution over $\mathcal{Z} = [0, 1]$. This corresponds to the case where $\dim Z < \dim \mathcal{X}$. The result is shown in Fig. 1a, where we can notice that the generated samples only reside on a self-entangled curve traversing the 2D manifold.



(a) 1→ 2 **(b) 2→ 1** **(c) 1→ 1**

Fig. 1 Manifold learning in 2D ambient space. In each subcaption, the number on the left of the arrow denotes the latent dimension of the generator while that on the right denotes the dimension of the target manifold.

What will happen if $\dim Z > \dim \mathcal{X}$, *i.e.*, if we approximate a curve with a surface? To investigate this, we build an 1D manifold immersed in $\mathbb{R}^2$ via the transformation $\phi(t) = (t - 0.5, t^3)$, then sample t uniformly from $[0, 1]$ and apply ϕ to obtain 100 data points on this 1D curve. After training an EGAN whose latent distribution is the uniform distribution over $\mathcal{Z} = [0, 1]^2$, as displayerd in Fig. 1b the generator can produce samples in the vicinity of the target 1D manifold and fit its shape well, yet because $\text{rank}\nabla g$ is almost everywhere equal to $\dim Z$, the image $g([0, 1]^2)$ cannot collapse into a single dimension, thus looks more like a narrow ribbon overlapping the 1D curve.

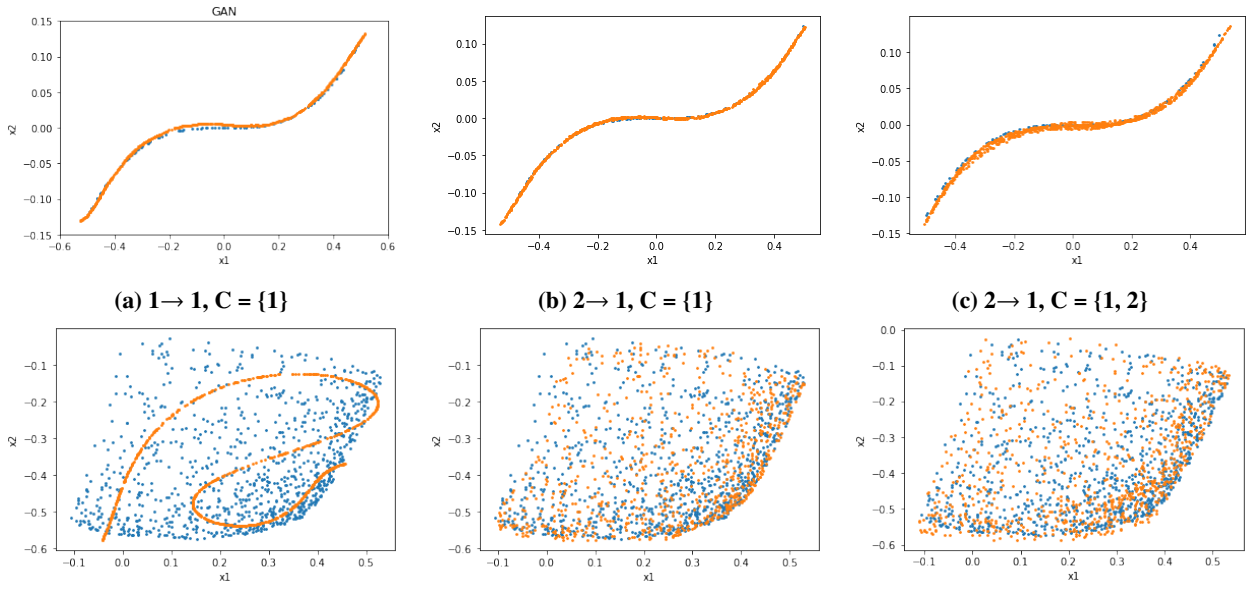
We may then expect the result to further improve when $\dim Z = \dim \mathcal{X}$, *i.e.*, when we fit a curve with a curve, yet it turns out to be too optimistic. The problem is that an ordinary ANN generator is not necessarily a globally injective function. This, together with the local immersion theorem, implies that in this 1D latent space case, the image of the generator may resemble a curve folding and intersecting itself (thus loses injectivity), which can be observed in Fig. 1c. Though we are fitting a curve with a curve, the result is even less ideal than the previous $\dim Z > \dim \mathcal{X}$ case, given that there exists some regions neighboring the target manifold that cannot be covered by this curve-generator whose image is of measure zero in the 2D ambient space, but can be covered if we use a surface-generator.

Therefore, these experiment results along with the differential topology theories suggest that when learning a manifold $\mathcal{X}$ immersed in a high dimensional ambient space, if there is no special regularization applied to the generator, we should choose a generator of latent dimension $\dim Z$ greater than $\dim \mathcal{X}$ for the best covering.

B. Effect of Mutual Information and Sinkhorn Dependency Measure Maximization

Since the maximization of mutual information and Sinkhorn dependency measure have similar effect in practice, for succinctness we only show the result of SDM maximization in this section and focus more on the effect itself. As a starter, for the instances "1 → 1" (Fig. 1c) and "1 → 2" (Fig. 1a) in §VI.A, we keep the rest of the experiment

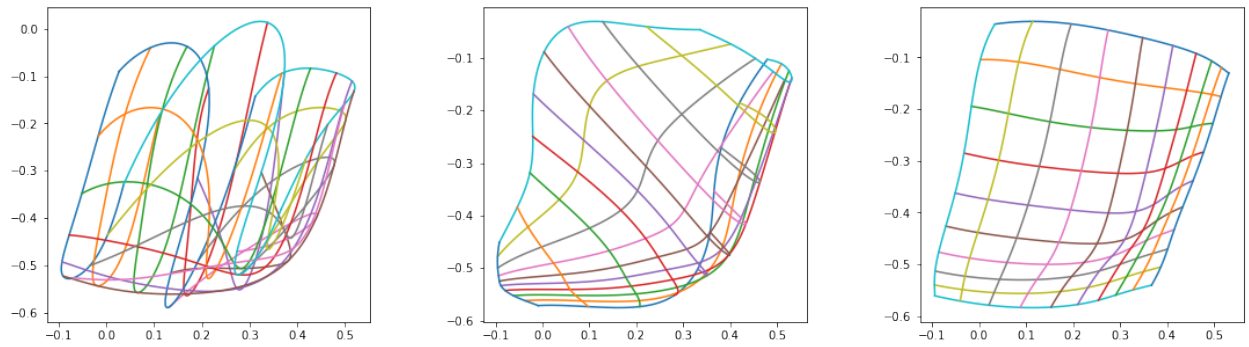
procedure unchanged and only in addition introduce SDM maximization for the first latent dimension when training EGAN, with the weight γ set to 0.1. The results are shown in Fig. 2a and 2d. We can see that unlike their predecessors, the curve—the image of the generator—is no longer entangled, which suggests the global injectivity of the generator over the latent space. This consequence can be loosely inferred from the property of mutual information, that for a random variable Y, $I(Y; g(Y))$ is only maximized when g is injective, although this statement only holds for discrete case and its extrapolation to SDM still remains to be verified rigorously. In addition, we can notice in Fig. 2a that due to the absence of self-entanglement, the SDM regularized learning result resembles the target manifold more than the one without regularization in Fig. 1c. This shows that SDM maximization, although not designed to drive p_g to p_r, may inadvertently improve manifold learning. SDM maximization also appears to improve the learning result for the case $2 \rightarrow 1$. Comparing Fig. 2b and Fig. 2c with Fig. 1b, we can see that g's "ribbon" image is now narrower and closer to the target manifold than before, which suggests a smaller $\mathrm{Im}\, g$. Informally speaking, this is because in both cases, one latent dimension undergoing SDM maximization learns to sufficiently align itself with the 1D curve, just like in Fig. 2a, so that the other latent dimension can focus on narrowing itself down. Although this is not illustrated here, it can be easily verified with the latent contour plot method below.



(a) 1→ 1, C = {1} **(b) 2→ 1, C = {1}** **(c) 2→ 1, C = {1, 2}**

(d) 1→ 2, C = {1} **(e) 2→ 2, C = {1}** **(f) 2→ 2, C = {1, 2}**

Fig. 2 Manifold learning in 2D ambient space. In each subcaption, the first two numbers follow the same denotion as in Fig. 1, whereas the numbers in the bracket denote the latent dimensions chosen as latent code.

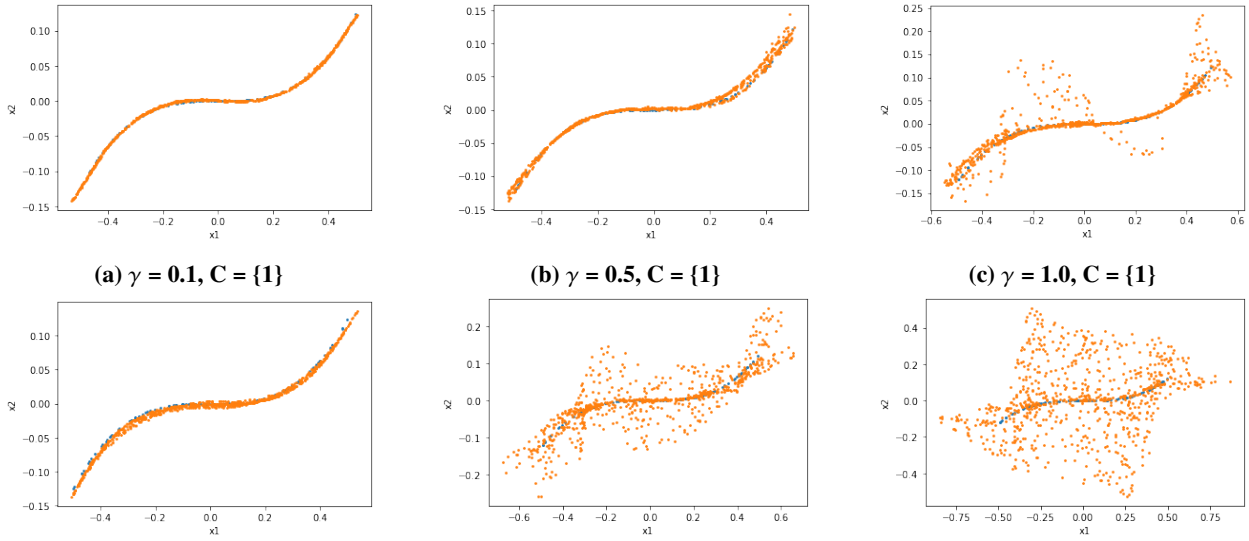


(a) 2→ 2, C = {∅} **(b) 2→ 2, C = {1}** **(c) 2→ 2, C = {1, 2}**

Fig. 3 Latent contour plot of the generators in the 2D manifold learning example.

It is then tempting to assume that a generator of higher dimensional latent space tends to be injective along each latent code dimension c_i undergoing SDM maximization, *i.e.*, $g(c, n)$ becomes an injective function over c_i if we keep the rest of the latent dimensions constant. It turns out to be a plausible hypothesis, at least through the lens of our *Latent Contour Plot* visualization in 2D setting presented in Fig. 3, the curves on which is produced by alternately varying each latent code dimension from 0 to 1 consistently while fixing the other latent dimensions at different values. If $g(c, n)$ is not injective along c_i, we can expect some of its corresponding curves to have self-intersection. This injection-encouraging characteristic of SDM maximization can indeed be observed in Fig. 3 clearly, for which we perform SDM maximization on none, only one and all the two latent dimensions in the 2D surface learning problem. Though the generators trained as such can all generate samples resembling the target distribution, as demonstrated in Fig. 2e and Fig. 2f, the underlying structures of the generators' images shown in Fig 3 are different, with the non-regularized generator having the most chaotic and self-intersected image (Fig. 3a), and the fully regularized one perching on the other end of the spectrum (Fig. 3c), having no sign of self-intersection and hence implying injectivity over $\mathcal{Z}$.

The effect of the SDM maximization weight γ is also worth investigating. To study this intuitively, we increase γ from 0.1 to 0.5 then 1 and plot the EGANs' learning results on the previous $2 \rightarrow 1$ manifold learning problem. In Fig. 4 we can see that as γ increases, in both the partial (first row of Fig. 4) and full (second row of Fig. 4) SDM maximization cases the image of g expands itself, leading to more generated samples deviating from the target manifold. We can also observe that the generator with all two latent dimensions being SDM maximized tends to have its image spread out more than the one with only a single latent dimension undergoing SDM maximization. This suggests the enforcement of injectivity is more of a byproduct of the SDM maximization process that analogously *stretches* thus unfolds the image of the generator, and this process needs to be carefully weighted against the Sinkhorn divergence minimization to mitigate its distortion effect on manifold learning. Moreover, we should keep the latent code space C's dimension not greater than the target manifold $\mathcal{X}$'s dimension to make the learning result less sensitive to the variation of γ.



(a) γ **= 0.1, C = {1}** **(b)** γ **= 0.5, C = {1}** **(c)** γ **= 1.0, C = {1}**

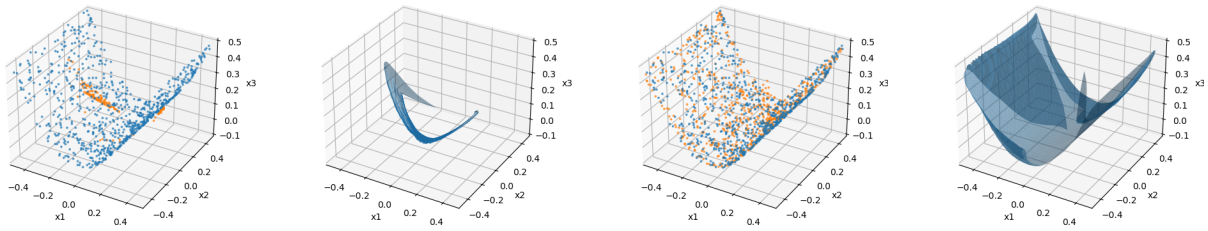
(d) γ **= 0.1, C = {1, 2}** **(e)** γ **= 0.5, C = {1, 2}** **(f)** γ **= 1.0, C = {1, 2}**

Fig. 4 SDM maximization with increasing weight γ on $2 \rightarrow 1$ manifold learning problem.

In summary, the low dimensional results show that SDM maximization (and mutual information maximization) can not only make $g(c, n)$ injective, but also consequently improve $\operatorname{Im} g$'s alignment with $\mathcal{X}$. The injectivity implies that we will encounter less optima when optimizing over $\mathcal{Z}$ through an SDM maximized g, which could simplify the optimization. However, we should carefully counterbalance this process and make the latent code space's dimension $\dim C$ not greater than the target manifold's dimension $\dim \mathcal{X}$ to alleviate its unwanted effect on manifold learning. The latter hint calls for a practical intrinsic dimension estimation technique, which shall be demonstrated in §VII.B.

C. Mode Collapse and Its Implication for Manifold Learning

In this experiment we demonstrate the predicament of mode collapse that vanilla GAN incurs and its detriment to learning a low dimensional target manifold $\mathcal{X}$ embedded in high dimensional X, *i.e.*, when $\mathcal{X}$ is of zero measure in X, which is analogous to the circumstance of airfoil manifold learning. The target manifold is a 2D surface in the 3D ambient space X, defined as the image of the map $f : [-0.5, 0.5] \times [-0.5, 0.5] \to X$ where $f(z_1, z_2) = [z_1, z_2, 2z_1^2]^\top$, and we sample points on this manifold (blue dots in Fig. 5) by transforming points uniformly sampled from $\mathcal{Z}$ with f. Two generators of latent dimension 2 and of the same hyperparameters and initializations are then trained by vanilla GAN and EGAN respectively for the same number of iterations to learn this 2D manifold, without MI and SDM maximization. Their latent distributions are identically the uniform distribution over $\mathcal{Z} = [0, 1]^2$.



(a) VGAN samples **(b) VGAN image** **(c) EGAN samples** **(d) EGAN image**

Fig. 5 The effect of mode collapse on manifold learning result.

The results are displayed in Fig. 5, where we can notice in Fig. 5a that VGAN suffers from mode collapse, with its generations (orange dots) concentrate only in a small area close to $\mathcal{X}$, whereas EGAN's generations (orange dots) in Fig. 5c appear to be sprayed over the entire $\mathcal{X}$, suggesting the absence of mode collapse. If we illustrate the image of these two generators, as in Fig. 5b and Fig. 5d, we can readily recognize their stark difference. The generator trained by VGAN can only cover a small portion of the target manifold $\mathcal{X}$ compared to the satisfying image of EGAN, which indicates that if after manifold learning, we optimize over $\mathcal{Z}$ through the VGAN generator, it is very likely that we cannot obtain optimal $x^\star$ under many boundary conditions. Although it might be possible to fine-tune vanilla GAN to mitigate this problem, the ideal combination of hyperparameters could take enormous effort to find. Therefore, this result bolsters our preference for EGAN over vanilla GAN in optimization applications given its robustness against mode collapse and the consequent improvement in the manifold learning result.

VII. Experiment: Airfoil Manifold Learning

In this final experiment we learn the airfoil manifold from the UIUC airfoil database*, which is composed of nearly 1600 valid airfoil designs. We first follow the same interpolation process introduced in our previous work [11] to unify these inconsistent airfoil representations with B-splines, then draw discrete data points from the B-splines as the final airfoil representation, with the sampling density determined by the regional curvature. The preprocessed airfoil samples are presented in Figure 6.

Fig. 6 Real airfoil samples from the UIUC airfoil dataset.

*http://m-selig.ae.illinois.edu/ads/coord_database.html

A. Performance of Entropic BézierGAN

The same Bézier curve generator as in [11] is employed to approximate the airfoil manifold. Thanks to the participation of Sinkhorn divergence and Sinkhorn algorithm, the auxiliary discriminator for the vanilla minimax game and the mutual information maximization can now be fully discarded. Compared with BézierGAN, this overhaul in turn reduces the running time of each training iteration by 3 times, while the number of iterations required to achieve a superior performance is at least 2 times less, thus in total the entropic BézierGAN is at least 6 times faster to train. Figure 7 shows two slices of the learned airfoil manifold with linear interpolation in the latent space. All the samples look realistic visually.

Quantitatively, the new entropic BézierGAN surpasses its predecessor in terms of most of the metrics in Table 1, while having comparable performance in terms of the rest. Its MMD [37] is significantly lower than that of its vanilla counterpart, suggesting less discrepancy between p_g and p_r. However, because of the two-fold role of g as introduced in the end of §II.B, we can hardly conclude whether to attribute this improvement more to the better alignment between $\mathrm{Im}\, g$ and $\mathcal{X}$, or to an improved local deformation of g, but only optimistically favor the former assumption in light of the result in §VI.C. The higher latent space consistency [39] suggests that points further away in the latent space tends to be mapped to more different airfoils, which in a sense reflects the improvement of g's injectivity. The increase in relative diversity [39] indicates that entropic BézierGAN can generate a wider variety of airfoils than vanilla BézierGAN, which ideally could result from an $\mathrm{Im}\, g$ covering more space in X. The relative variance of difference (RVOD) [45] measures the smoothness of the generated Bézier curves. Although now lower, it is still very close to the original BézierGAN's result, and the realistic airfoil contours in Fig. 7 support our disregard of this tiny discrepancy.

Table 1 Metrics comparison between original BézierGAN and entropic BézierGAN

Model	MMD [37]	Latent Space Consistency [39]	Relative Diversity [39]	RVOD [45]
Original	0.1856±0.0007	0.9432±0.0025	1.0455±0.0144	**1.0291±0.0004**
Entropic	**0.0896±0.0015**	**0.9878±0.0005**	**1.3264±0.0196**	0.9702±0.0005

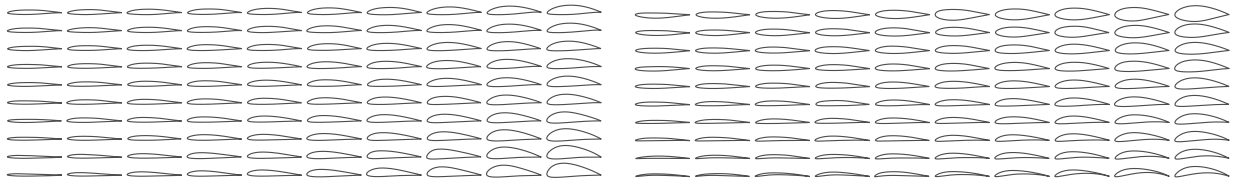


Fig. 7 Synthesized airfoils in the three dimensional latent code space.

B. Effect of Latent Dimension on Airfoil Manifold Learning

We have seen in the previous section that the weight of SDM maximization needs to be fine tuned, especially when the dimension of the latent code space is greater than that of the target manifold. Therefore, determining the intrinsic dimension of the latent manifold beforehand can help us select the right latent code and latent noise dimension, significantly facilitate the manifold learning process and improve the learning result.

We report results for the estimation of intrinsic dimensionality in Figure 8. For the MLE estimator, we observe the the estimates concentrate around $d = 4$ for values of $k \geq 10$. This convergence suggests to us that the true estimate lies in this range. For the TwoNN estimator, estimates start around $d = 4$ but then diverge to approximately 2.5. This fluctuation is large relative to the scale of the initial and other estimates, and therefore must be interpreted with caution. The GeoMLE estimator converges smoothly to between $d = 5$ and $d = 6$. All told, the three estimators suggest the true estimate is somewhere between 2 and 6, with somewhat greater probability around 4.

We then inspect the performance of EBGANs assigned with different latent code dimensions from 2 to 5 to verify the accuracy of these techniques. The variation of different metrics against the latent code dimension is illustrated in Fig. 9, where we can see that as the dimension transcends our largest MLE estimation 4, the performance of the EBGAN shifts dramatically in terms of all four metrics. This reminds us of the behavior of EGAN in Fig. 4 that the generator's performance suddenly deteriorates when $\dim C$ rises beyond $\dim \mathcal{X}$. Indeed, if we look into the slices of the latent code

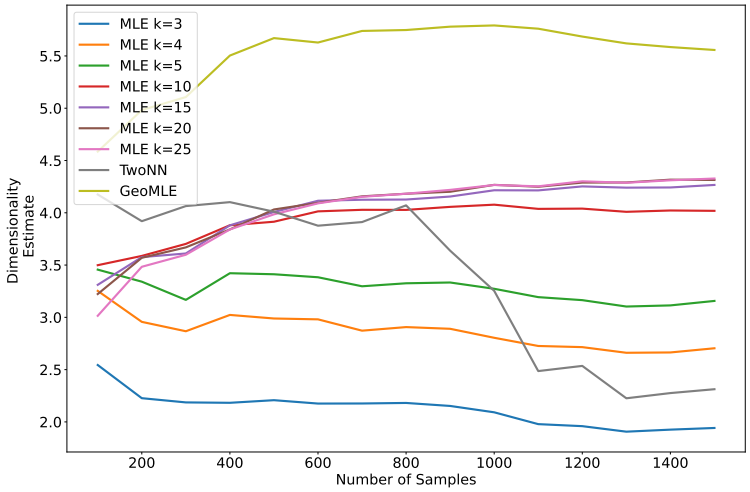
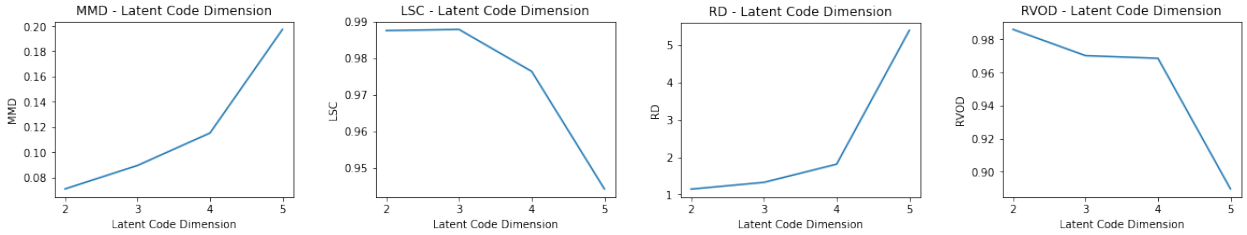


Fig. 8 Dimensionality estimation of UIUC data with MLE, TwoNN, and GeoMLE methods.



(a) MMD **(b) LSC** **(c) RD** **(d) RVOD**

Fig. 9 The effect of latent code dimension on UIUC airfoil manifold learning result, through the lens of metrics.

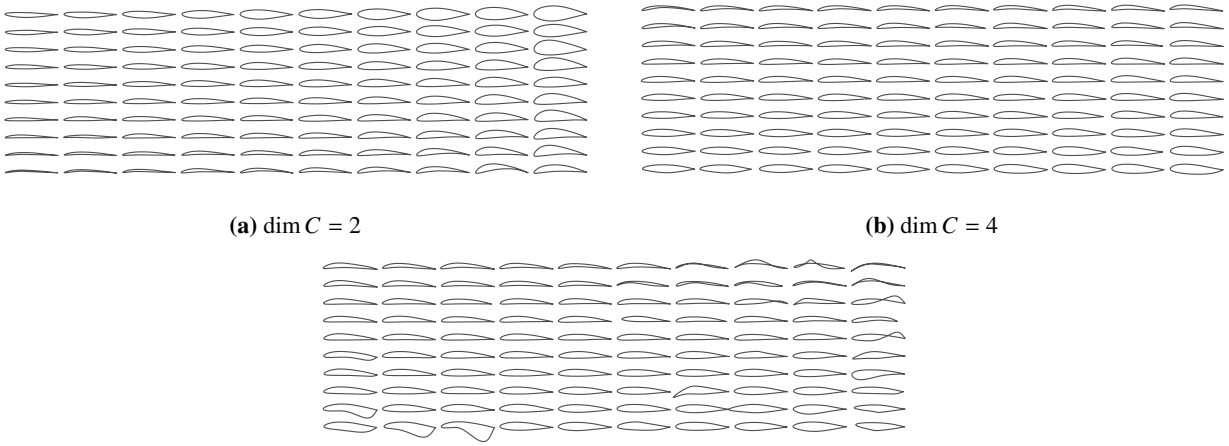
space for these four $\dim C$ setups, we can perceive an abrupt emergence of unrealistic airfoils in the samples produced by the generator with $\dim C = 5$, as shown in Fig. 10c compared to Fig. 10a, Fig. 10b and Fig. 7. This is reminiscent of the large amount of points in Fig. 4e that are far away from the target manifold, in stark contrast with the situation in Fig. 4b. Therefore, based on our experience in Fig. 4 that a dimension mismatch $\dim C > \dim \mathcal{X}$ promptly induces aberrant samples outside the manifold, we can speculate that the dimension of the UIUC airfoil manifold should be lower than 5, which agrees well with our MLE estimations, and we ought to keep $\dim C$ under this value for the best manifold learning result. The efficacy of MLE intrinsic dimension estimation technique is thereby confirmed.

VIII. Discussion and Conclusion

In this work, we overhauled BézierGAN into entropic BézierGAN, based on the recent development in computational optimal transport. Consequently, the airfoil manifold learning process is accelerated by at least 6 times, while the learning result is also considerably improved through the lens of a variety of metrics. On top of that, with the help of low dimensional toy problems, we probed into and elaborated on the relationship between the latent dimension of the generator and the target manifold's dimension, the injectivity-encouraging effect of Sinkhorn dependency measure (and mutual information) maximization and the detrimental influence of mode collapse plaguing vanilla GANs, then came up with the idea of employing MLE intrinsic dimension estimation methodology to approximate the target manifold's dimension beforehand to help select proper latent code and noise dimension, which can improve the learning result while sparing the hassle of brute-force hyperparameter tuning.

References

[1] Samareh, J. A., "Survey of shape parameterization techniques for high-fidelity multidisciplinary shape optimization," *AIAA journal*, Vol. 39, No. 5, 2001, pp. 877–884.



(a) $\dim C = 2$ **(b)** $\dim C = 4$

(c) $\dim C = 5$

Fig. 10 Slices of the latent code space for $\dim C = 2, 4, 5$**.**

[2] Bellman, R., *Dynamic Programming*, Princeton Landmarks in Mathematics and Physics, Princeton University Press, 2010.

[3] Lukaczyk, T. W., Constantine, P., Palacios, F., and Alonso, J. J., "Active subspaces for shape optimization," *10th AIAA multidisciplinary design optimization conference*, 2014, p. 1171.

[4] Berguin, S. H., and Mavris, D. N., "Dimensionality Reduction In Aerodynamic Design Using Principal Component Analysis With Gradient Information," *10th AIAA Multidisciplinary Design Optimization Conference*, 2014, p. 0112.

[5] Diez, M., Serani, A., Campana, E. F., Volpi, S., and Stern, F., "Design space dimensionality reduction for single-and multi-disciplinary shape optimization," *17th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, 2016, p. 4295.

[6] Cinquegrana, D., and Iuliano, E., "Efficient global optimization of a transonic wing with geometric data reduction," *35th AIAA Applied Aerodynamics Conference*, 2017, p. 3057.

[7] Viswanath, A., Forrester, A., and Keane, A., "Constrained design optimization using generative topographic mapping," *AIAA journal*, Vol. 52, No. 5, 2014, pp. 1010–1023.

[8] Yasong, Q., Junqiang, B., Nan, L., and Chen, W., "Global aerodynamic design optimization based on data dimensionality reduction," *Chinese Journal of Aeronautics*, Vol. 31, No. 4, 2018, pp. 643–659.

[9] Li, J., Bouhlel, M. A., and Martins, J. R., "Data-based approach for fast airfoil analysis and optimization," *AIAA Journal*, Vol. 57, No. 2, 2019, pp. 581–596.

[10] Poole, D. J., Allen, C. B., and Rendall, T., "Efficient Aero-Structural Wing Optimization Using Compact Aerofoil Decomposition," *AIAA Scitech 2019 Forum*, 2019, p. 1701.

[11] Chen, W., Chiu, K., and Fuge, M. D., "Airfoil Design Parameterization and Optimization Using Bézier Generative Adversarial Networks," *AIAA Journal*, Vol. 58, No. 11, 2020, pp. 4723–4735.

[12] Boncoraglio, G., and Farhat, C., "Active Manifold and Model Reduction for Multidisciplinary Analysis and Optimization," *AIAA Scitech 2021 Forum*, 2021, p. 1694.

[13] Fefferman, C., Mitter, S., and Narayanan, H., "Testing the manifold hypothesis," *Journal of the American Mathematical Society*, Vol. 29, No. 4, 2016, pp. 983–1049.

[14] Chen, W., Fuge, M., and Chazan, J., "Design manifolds capture the intrinsic complexity and dimension of design spaces," *Journal of Mechanical Design*, Vol. 139, No. 5, 2017.

[15] Metz, L., Poole, B., Pfau, D., and Sohl-Dickstein, J., "Unrolled generative adversarial networks," *arXiv preprint arXiv:1611.02163*, 2016.

[16] Peyré, G., Cuturi, M., et al., "Computational optimal transport: With applications to data science," *Foundations and Trends® in Machine Learning*, Vol. 11, No. 5-6, 2019, pp. 355–607.

[17] Chen, X., Duan, Y., Houthooft, R., Schulman, J., Sutskever, I., and Abbeel, P., "Infogan: Interpretable representation learning by information maximizing generative adversarial nets," *arXiv preprint arXiv:1606.03657*, 2016.

[18] Pope, P., Zhu, C., Abdelkader, A., Goldblum, M., and Goldstein, T., "The Intrinsic Dimension of Images and Its Impact on Learning," *arXiv preprint arXiv:2104.08894*, 2021.

[19] Bishop, C. M., *Pattern recognition and machine learning, 5th Edition*, Information science and statistics, Springer, 2007. URL https://www.worldcat.org/oclc/71008143.

[20] Goodfellow, I., Bengio, Y., and Courville, A., *Deep Learning*, MIT Press, 2016. http://www.deeplearningbook.org.

[21] Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y., "Generative adversarial networks," *arXiv preprint arXiv:1406.2661*, 2014.

[22] Nowozin, S., Cseke, B., and Tomioka, R., "f-GAN: Training Generative Neural Samplers using Variational Divergence Minimization," *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, edited by D. D. Lee, M. Sugiyama, U. von Luxburg, I. Guyon, and R. Garnett, 2016, pp. 271–279. URL https://proceedings.neurips.cc/paper/2016/hash/cedebb6e872f539bef8c3f919874e9d7-Abstract.html.

[23] Mao, X., Li, Q., Xie, H., Lau, R. Y. K., Wang, Z., and Smolley, S. P., "Least Squares Generative Adversarial Networks," *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*, IEEE Computer Society, 2017, pp. 2813–2821. https://doi.org/10.1109/ICCV.2017.304, URL https://doi.org/10.1109/ICCV.2017.304.

[24] Arjovsky, M., Chintala, S., and Bottou, L., "Wasserstein generative adversarial networks," *International conference on machine learning*, PMLR, 2017, pp. 214–223.

[25] Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., and Courville, A., "Improved training of wasserstein gans," *arXiv preprint arXiv:1704.00028*, 2017.

[26] Kingma, D. P., and Welling, M., "Auto-Encoding Variational Bayes," *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, edited by Y. Bengio and Y. LeCun, 2014. URL http://arxiv.org/abs/1312.6114.

[27] Kingma, D. P., Salimans, T., and Welling, M., "Improving Variational Inference with Inverse Autoregressive Flow," *CoRR*, Vol. abs/1606.04934, 2016. URL http://arxiv.org/abs/1606.04934.

[28] Dinh, L., Krueger, D., and Bengio, Y., "NICE: Non-linear Independent Components Estimation," *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Workshop Track Proceedings*, edited by Y. Bengio and Y. LeCun, 2015. URL http://arxiv.org/abs/1410.8516.

[29] Dinh, L., Sohl-Dickstein, J., and Bengio, S., "Density estimation using Real NVP," *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*, OpenReview.net, 2017. URL https://openreview.net/forum?id=HkpbnH9lx.

[30] Kingma, D. P., and Dhariwal, P., "Glow: Generative Flow with Invertible 1x1 Convolutions," *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, edited by S. Bengio, H. M. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, 2018, pp. 10236–10245. URL https://proceedings.neurips.cc/paper/2018/hash/d139db6a236200b21cc7f752979132d0-Abstract.html.

[31] Ben-Israel, A., "The change-of-variables formula using matrix volume," *SIAM Journal on Matrix Analysis and Applications*, Vol. 21, No. 1, 1999, pp. 300–312.

[32] Arjovsky, M., and Bottou, L., "Towards principled methods for training generative adversarial networks," *arXiv preprint arXiv:1701.04862*, 2017.

[33] Genevay, A., Peyré, G., and Cuturi, M., "Learning generative models with sinkhorn divergences," *International Conference on Artificial Intelligence and Statistics*, PMLR, 2018, pp. 1608–1617.

[34] Sanjabi, M., Ba, J., Razaviyayn, M., and Lee, J. D., "On the convergence and robustness of training gans with regularized optimal transport," *arXiv preprint arXiv:1802.08249*, 2018.

[35] Balaji, Y., Hassani, H., Chellappa, R., and Feizi, S., "Entropic GANs meet VAEs: A statistical approach to compute sample likelihoods in GANs," *International Conference on Machine Learning*, PMLR, 2019, pp. 414–423.

[36] Feydy, J., Séjourné, T., Vialard, F.-X., Amari, S.-i., Trouvé, A., and Peyré, G., "Interpolating between optimal transport and MMD using Sinkhorn divergences," *The 22nd International Conference on Artificial Intelligence and Statistics*, PMLR, 2019, pp. 2681–2690.

[37] Smola, A., Gretton, A., Song, L., and Schölkopf, B., "A Hilbert space embedding for distributions," *International Conference on Algorithmic Learning Theory*, Springer, 2007, pp. 13–31.

[38] Ozair, S., Lynch, C., Bengio, Y., Oord, A. v. d., Levine, S., and Sermanet, P., "Wasserstein dependency measure for representation learning," *arXiv preprint arXiv:1903.11780*, 2019.

[39] Chen, W., and Fuge, M., "Synthesizing designs with interpart dependencies using hierarchical generative adversarial networks," *Journal of Mechanical Design*, Vol. 141, No. 11, 2019.

[40] Levina, E., and Bickel, P. J., "Maximum likelihood estimation of intrinsic dimension," *Advances in neural information processing systems*, 2005, pp. 777–784.

[41] MacKay, D. J., and Ghahramani, Z., "Comments on'Maximum likelihood estimation of intrinsic dimension'by E. Levina and P. Bickel (2004)," *The Inference Group Website, Cavendish Laboratory, Cambridge University*, 2005.

[42] Facco, E., d'Errico, M., Rodriguez, A., and Laio, A., "Estimating the intrinsic dimension of datasets by a minimal neighborhood information," *Scientific Reports*, Vol. 7, No. 1, 2017, p. 12140.

[43] Gomtsyan, M., Mokrov, N., Panov, M., and Yanovich, Y., "Geometry-Aware Maximum Likelihood Estimation of Intrinsic Dimension," *Asian Conference on Machine Learning*, 2019, pp. 1126–1141.

[44] Guillemin, V., and Pollack, A., *Differential topology*, Vol. 370, American Mathematical Soc., 2010.

[45] Chen, W., and Fuge, M., "B\'ezierGAN: Automatic Generation of Smooth Curves from Interpretable Low-Dimensional Parameters," *arXiv preprint arXiv:1808.08871*, 2018.