Spider GAN: Leveraging Friendly Neighbors to Accelerate GAN Training

Siddarth Asokan*
Robert Bosch Center for Cyber-Physical Systems
Indian Institute of Science
Bengaluru - 50012, India
siddartha@iisc.ac.in

Chandra Sekhar Seelamantula
Department of Electrical Engineering
Indian Institute of Science
Bengaluru - 50012, India
css@iisc.ac.in

Abstract

Training Generative adversarial networks (GANs) stably is a challenging task. The generator in GANs transform noise vectors, typically Gaussian distributed, into realistic data such as images. In this paper, we propose a novel approach for training GANs with images as inputs, but without enforcing any pairwise constraints. The intuition is that images are more structured than noise, which the generator can leverage to learn a more robust transformation. The process can be made efficient by identifying closely related datasets, or a "friendly neighborhood" of the target distribution, inspiring the moniker, Spider GAN. To define friendly neighborhoods leveraging proximity between datasets, we propose a new measure called the signed inception distance (SID), inspired by the polyharmonic kernel. We show that the Spider GAN formulation results in faster convergence, as the generator can discover correspondence even between seemingly unrelated datasets, for instance, between Tiny-ImageNet and CelebA faces. Further, we demonstrate cascading Spider GAN, where the output distribution from a pre-trained GAN generator is used as the input to the subsequent network. Effectively, transporting one distribution to another in a cascaded fashion until the target is learnt – a new flavor of transfer learning. We demonstrate the efficacy of the Spider approach on DCGAN, conditional GAN, PG-GAN, StyleGAN2 and StyleGAN3. The proposed approach achieves state-of-the-art Fréchet inception distance (FID) values, with one-fifth of the training iterations, in comparison to their baseline counterparts on high-resolution small datasets such as MetFaces, Ukiyo-E Faces and AFHQ-Cats.

1. Introduction

Generative adversarial networks (GANs) [1] are designed to model the underlying distribution of a target dataset (with underlying distribution p_d) through a *min-max* optimization between the generator G and the discriminator D networks. The generator transforms an input $\boldsymbol{z} \sim p_z$, typically Gaussian or uniform distributed, into a generated sample $G(\boldsymbol{z}) \sim p_g$. The discriminator is trained to classify samples drawn from p_g or p_d as real or fake. The optimal generator is the one that outputs images that confuse the discriminator.

Inputs to the GAN generator: The input distribution plays a definitive role in the quality of GAN output. Low-dimensional latent vectors have been shown to help disentangle the representations and control features of the target being learnt [2, 3]. Prior work on optimizing the latent distribution in GANs has been motivated by the need to improve the quality of interpolated images. Several works have considered replacing the Gaussian prior with Gaussian mixtures, Gamma, non-parametric distributions, etc [4–9]. Alternatively, the GAN generator can be trained with the latent-space distribution of the target dataset, as learnt by variational autoencoders [10, 11]. However, such approaches are not in conformity with the low-dimensional manifold structure of real data. Khayatkhoei *et al.* [12] attributed the poor quality of the interpolates to the disjoint structure of data distribution in high-dimensions, which motivates the need for an informed choice of the input distribution.

GANs and image-to-image translation: GANs that accept images as input fall under the umbrella of *image translation*. Here, the task is to modify particular features of an image, either within domain (style transfer) or across domains (domain adaptation). Examples for in-domain translation include changing aspects of face images, such as the expression, gender, accessories, etc. [13–15], or modifying the illumination or seasonal characteristics of natural scenes [16]. On the other hand, *domain adaptation tasks* aim at transforming the image from one style to another. Common applications include simulation to real-world translation [17–20], or translating images across styles of artwork [21–23]. While the supervised Pix2Pix framework [22] originally proposed training GANs with pairs of images drawn from the source and target domains, semi-supervised and unsupervised ex-

*Siddarth Asokan is funded by the Qualcomm Innovation Fellowship, and the Robert Bosch Center for Cyber-Physical Systems Ph.D. Fellowship.

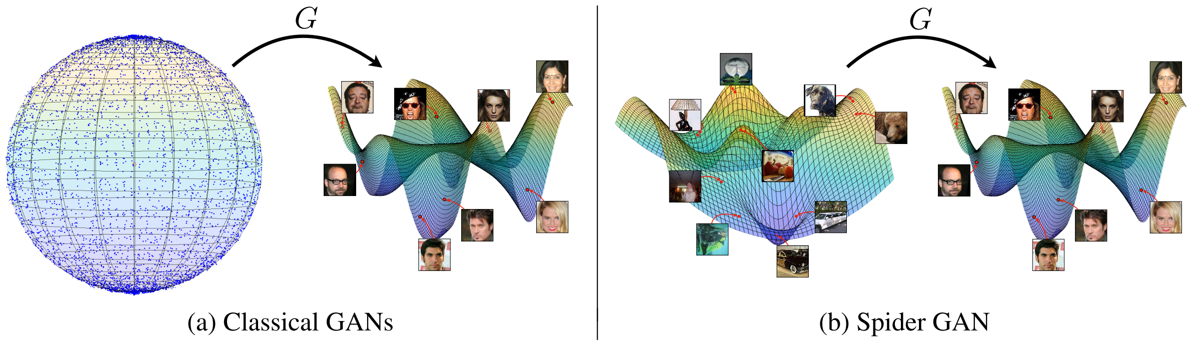

Figure 1. (Color online) A comparison of design philosophies of the standard GANs and Spider GAN. (a) A prototypical GAN transforms high-dimensional Gaussian data, which is concentrated at the surface of hyperspheres in n-D, into an image distribution comprising a union of low-dimensional manifolds embedded in a higher-dimensional space. (b) The Spider GAN generator aims to learn a simpler transformation between two closely related data manifolds in an unconstrained manner, thereby accelerating convergence.

tensions [23–28] tackle the problem in an unpaired setting, and introduce modifications such as cycle-consistenty or the addition of regularization functionals to the GAN loss to maintain a measure of consistency between images. Existing domain-adaptation GANs [29, 30] enforce cross-domain consistency to retain visual similarity. Ultimately, these approaches rely on enforcing some form of coupling between the source and the target via feature-space mapping.

2. The Proposed Approach: Spider GAN

We propose the Spider GAN formulation motivated by the low-dimensional disconnected manifold structure of data [12, 31–33]. Spider GANs lie at the cross-roads between classical GANs and image-translation GANs. As opposed to optimizing the latent parametric prior, we hypothesize that providing the generator with closely related image source datasets, (dubbed the *friendly neighborhood*, leading to the moniker *Spider GAN*) will result in superior convergence of the GAN. Unlike image translation tasks, the Spider GAN generator is agnostic to individual input-image features, and is allowed to *discover* implicit structure in the mapping from the source distribution to the target. Figure 1 depicts the design philosophy of Spider GAN juxtaposed with the classical GAN training approach.

The choice of the *input dataset* affects the generator's ability to learn a stable and accurate mapping. Intuitively, if the GAN has to be trained to learn the distribution of *street view house numbers* (SVHN) [34], the MNIST [35] dataset proves to be a better initialization of the input space than standard densities such as the uniform or Gaussian. It is a well known result that, for a given mean and variance, the Gaussian has maximum entropy, while for a given support (say, $[-1, 1]$ when training with re-normalized images), the uniform distribution has maximum entropy [36]. However, image datasets are highly structured, and possess lower entropy [37]. Therefore, one could interpret the generative modeling of images using GANs as effectively one of entropy minimization [13]. We argue that choosing a low entropy input distribution that is structurally closer to the target would lead to a more efficient generator transformation, thereby accelerating the training process. Existing image-translation approaches aim to maintain semantic information, for example, translating a specific instance of the digit '2' in the MNIST dataset to the SVHN style. However, the Spider GAN formulation neither enforces nor requires such constraints. Rather, it allows for an implicit structure in the source dataset to be used to learn the target efficiently. It is entirely possible for the *Trouser* class in Fashion-MNIST [38] to map to the digit '1' in MNIST due to structural similarity. Thus, the scope of Spider GAN is much wider than image translation.

2.1. Our Contributions

In Section 3, we discuss the central focus in Spider GANs: defining what constitutes a *friendly neighborhood*. Preliminary experiments suggest that, while the well known Fréchet inception distance (FID) [39] and kernel inception distance (KID) [40] are able to capture visual similarity, they are unable to quantify the diversity of samples in the underlying manifold. We therefore propose a novel distance measure to evaluate the input to GANs, one that is motivated by electrostatic potential fields and charge neutralization between the (positively charged) target data samples and (negatively charged) generator samples [41, 42], named *signed inception distance* (SID) (Section 3.1). An implementation of SID atop the Clean-FID [43] backbone is available at `https://github.com/DarthSid95/clean-sid`. We identify *friendly neighborhoods* for multiple classes of standard image datasets such as MNIST, Fashion MNIST, SVHN, CIFAR-10 [44], Tiny-ImageNet [45], LSUN-Churches [46], CelebA [47], and Ukiyo-E Faces [48]. We present experimental validation on training the *Spider* variant of DCGAN [49] (Section 4) and show that it results in up to 30% improvement in terms of FID, KID and cumulative SID of the converged models. The *Spider* framework is lightweight and can be extended to any GAN architecture, which we demonstrate via class-conditional learning with the *Spider* variant of auxiliary classifier GANs (ACGANs) [50] (Section 4). The source code for Spider GANs built atop the DCGAN architecture are available at https://github.com/DarthSid95/SpiderDCGAN. We also present a novel approach to transfer learning using Spider GANs by feeding the output distribution of a pre-trained generator to the input of the subsequent stage (Section 5). Considering progressively growing GAN (PGGAN) [51] and StyleGAN [52–54] architectures, we show that the corresponding *Spider* variants achieve competitive FID scores in one-fifth of the training iterations on FFHQ [14] and AFHQ-Cats [30], while achieving state-of-the-art FID on high-resolution small-sized datasets such as Ukiyo-E Faces and MetFaces [53] (Section 5.1). The source code for implementing Spider StyleGANs is available at https://github.com/DarthSid95/SpiderStyleGAN.

2.2. Related Works

The choice of the input distribution in GANs determines the quality of images generated by feeding the generator interpolated points, which in turn is determined by the probability of the interpolated points lying on the manifold. High-dimensional Gaussian random vectors are concentrated on the surface of a hypersphere (*Gaussian annulus theorem* [55]), akin to a *soap bubble*, resulting in interpolated points that are less likely to lie on the manifold. Alternatives such as the Gamma [6] or Cauchy [7] prior result in superior performance over interpolated points, while Singh *et al.* [9] derive a non-parametric prior that minimized the divergence between the input and the midpoint distributions.

A well known result in high-dimensional data analysis is that structured datasets are embedded in a low-dimensional manifold with an *intrinsic dimensionality* ($n_{\mathfrak{D}}$) significantly lower than the ambient dimensionality n [37]. For instance, in MNIST, $n = 784$, while $n_{\mathfrak{D}} \approx 12$ [56]. Feng *et al.*[57] showed that the mismatch between $n_{\mathfrak{D}}$ of the generator input and its output adversely affects performance. Although in practice, estimating $n_{\mathfrak{D}}$ may not always be possible [12, 56, 58], these results justify picking input distributions that are structurally similar to the target. In instance-conditioned GANs [59], the target data is modeled as clusters on the data manifold to improve learning.

The philosophy of cascading Spider GAN generators runs in parallel to input optimization in transfer learning with GANs, such as Mine GAN [60] where *mining* networks are implemented that transform the input distribution of the GAN nonlinearly to learn the target samples better. Kerras *et al.* [53] showed that transfer learning improves the performance of GANs on small datasets, and observed empirically that transferring weights from models trained on visually diverse data lead to better performance of the target model.

3. Where is the Friendly Neighborhood?

We now consider various distance measures between datasets that can be used to identify the friendly neighborhood/source dataset in Spider GANs. While the most direct approach is to compare the intrinsic dimensions of the manifolds, such approaches are either computationally intensive [61], or do not scale with sample size [56, 58]. We observed that the friendly neighbors detected by such approach did not correlate well experimentally, and therefore, defer discussions on such methods to Appendix A.

Based on the approach advocated by Wang *et al.* [62] to identify pre-trained GAN networks for transfer learning, we initially considered FID and KID to identify *friendly neighbors*. We use the FID to measure the distance between the source (generator input) and the target data distributions. A source that has a lower FID is closer to the target and will serve as a better input to the generator. The first four columns of Table 1 present FID scores between the standard datasets we consider in this paper. The **first**, **second** and **third** *friendly neighbors* (color coded) of a target dataset are the source datasets with the lowest three FIDs. As observed from Table 1, a limitation is that the FID of a dataset with itself is not always zero, which is counterintuitive for a distance measure. In cases such as CIFAR-10 or Tiny-ImageNet, this is indicative of the variability in the dataset, and in Ukiyo-E Faces, this is due to limited availability of data samples, which has been shown to negatively affect FID estimation [40, 63]. FID satisfies *reciprocity*, *i.e.,* it identifies datasets as being mutually close to each other, such as CIFAR-10 and Tiny-ImageNet. However, preliminary experiments on training Spider GAN using FID to identify friendly neighbors showed that the relative diversity between datasets is not captured. Given a source, learning a less diverse target distribution is easier (cf. Section 4 and Appendix D.2). These issues are similar to the observations made by Kerras *et al.* [53] in the context of weight transfer. This can be understood via an example — fitting a multimodal target Gaussian having 10 modes would be easier with a 20-component source distribution than a 5-component one.

3.1. The Signed Inception Distance (SID)

Given the limitations of FID discussed above, we propose a novel signed distance for measuring the proximity between two distributions. The distance is "signed" in the sense that it can also take negative values. Further, it is not symmetric. The distance is also practical to compute because it is expressed in terms of the samples drawn from the distributions. The proposed distance draws inspiration from the improved precision-recall scores of GANs [64] and the potential-field interpretation in Coulomb GANs [41] and Poly-LSGAN [42]. Consider batches of samples drawn from distributions μ_p and μ_q, given by $\mathfrak{D}_p = \{\tilde{\boldsymbol{c}}_i\}_{i=1}^{N_p}$ and

Table 1. A comparison of FID and CSID_m between popular training datasets for $m = \lfloor \frac{n}{2} \rfloor$. The rows represent the source and the columns correspond to the target. The **first**, **second** and **third** *friendly neighbors* of the target are the sources with the three lowest FID, or lowest positive CSID values, respectively. CSID is superior to FID, as it assigns negative values to sources that are less diverse than the target. MNIST and Fashion-MNIST are shown in gray to denote scenarios where grayscale images are not valid sources for the color-image targets.

Source \ Target	FID (Source , Target)				CSID_m(Source ‖ Target)			
	MNIST	CIFAR-10	TinyImageNet	Ukiyo-E	MNIST	CIFAR-10	TinyImageNet	Ukiyo-E
MNIST	1.2491	258.246	264.250	398.280	0.1863	29.298	9.436	201.550
F-MNIST	**176.813**	188.367	197.057	387.049	**162.962**	19.051	-2.5571	191.010
SVHN	**236.707**	**168.615**	**189.133**	372.444	212.473	**34.534**	**21.668**	214.507
CIFAR-10	**259.045**	5.0724	**64.3941**	303.694	221.337	-0.1487	-7.109	198.991
TinyImageNet	264.309	**64.0312**	6.4854	**257.078**	230.916	**12.892**	0.6743	**197.447**
CelebA	360.773	303.490	**250.735**	**301.108**	**204.794**	**23.685**	**8.829**	**184.170**
Ukiyo-E	396.791	300.511	254.102	5.9137	250.226	39.793	**18.727**	0.5494
Church	350.708	**294.982**	254.991	**267.638**	**212.452**	-4.655	-23.115	**198.750**

$\mathfrak{D}_q = \{\boldsymbol{c}_j\}_{j=1}^{N_q}$, respectively. Given a test vector $\boldsymbol{x} \in \mathbb{R}^n$, consider the Coulomb GAN discriminator [41]:

$$f(\boldsymbol{x}) = \frac{1}{N_p} \sum_{\substack{i=1 \\ \tilde{\boldsymbol{c}}_i \sim \mu_p}}^{N_p} \Phi(\boldsymbol{x}, \tilde{\boldsymbol{c}}_i) - \frac{1}{N_q} \sum_{\substack{j=1 \\ \boldsymbol{c}_j \sim \mu_q}}^{N_q} \Phi(\boldsymbol{x}, \boldsymbol{c}_j), \tag{1}$$

where Φ is the polyharmonic kernel [42, 65]:

$$\Phi(\boldsymbol{x}, \boldsymbol{y}) = \kappa_{m,n} \begin{cases} \|\boldsymbol{x}-\boldsymbol{y}\|^{2m-n}, & \text{if } 2m-n<0 \text{ or } n \text{ is odd,} \\ \|\boldsymbol{x}-\boldsymbol{y}\|^{2m-n} \ln(\|\boldsymbol{x}-\boldsymbol{y}\|), & \text{if } 2m-n\geq 0 \text{ and } n \text{ is even,} \end{cases}$$

and $\kappa_{m,n}$ is a positive constant, given the order m and dimensionality n. The higher-order generalization gives us more flexibility and numerical stability in computation. We use $m \approx \lfloor \frac{n}{2} \rfloor$ as a stable choice, while ablation studies on choosing m are given in Appendix B.4

From the perspective of electrostatics, for $\mu_p = p_g$ and $\mu_q = p_d$, $f(\boldsymbol{x})$ in Equation (1) treats the target data as negative charges, and generator samples as positive charges. The quality of μ_p in approximating/matching μ_q is measurable by computing the effect of the net charge present in any chosen volume around the target μ_q on a test charge $\boldsymbol{x}$. Consider a hypercube $\mathcal{C}_{q,r}$ of side length r, centered around μ_q with test charges $\{\boldsymbol{x}_\ell\}_{\ell=1}^{M_{\boldsymbol{x}}}$, $\boldsymbol{x}_\ell \in \mathcal{C}_{q,r}$. To analyze the average behavior of target and generated samples in $\mathcal{C}_{q,r}$, we draw $\boldsymbol{x}_\ell$ uniformly within $\mathcal{C}_{q,r}$. We consider $N_p = N_q = N$ for simplicity. We now define the *signed distance* of μ_p from μ_q as the negative of $f(\boldsymbol{x})$, summed over a uniform sampling of points over $\mathcal{C}_{q,r}$, *i.e.* $\text{SD}_{m,r}(\mu_p \| \mu_q)$ is given by:

$$\frac{1}{NM_{\boldsymbol{x}}} \sum_{\substack{\ell=1 \\ \tilde{\boldsymbol{x}}_\ell \in \mathcal{C}_{q,r}}}^{M_{\boldsymbol{x}}} \left(\sum_{\substack{j=1 \\ \boldsymbol{c}_j \sim \mu_q}}^{N} \Phi(\boldsymbol{x}_\ell, \boldsymbol{c}_j) - \sum_{\substack{i=1 \\ \tilde{\boldsymbol{c}}_i \sim \mu_p}}^{N} \Phi(\boldsymbol{x}_\ell, \tilde{\boldsymbol{c}}_i) \right). \tag{2}$$

Similar to the improved precision and recall (IPR) metrics, $\text{SD}_{m,r}(\mu_p \| \mu_q)$ is asymmetrical, *i.e.,* $\text{SD}_{m,r}(\mu_p \| \mu_q) \neq \text{SD}_{m,r}(\mu_q \| \mu_p)$. When $\text{SD}_{m,r}(\mu_p \| \mu_q) < 0$, on the average, samples from μ_q are relative more spread out than those drawn from μ_p with respect to $\mathcal{C}_{q,r}$, and vice versa. When $\mu_p = \mu_q$, we have $\text{SD}_{m,r}(\mu_p \| \mu_q) \approx 0$. Illustrations of these three scenarios are provided in Appendix B.3.

In practice, similar to the standard GAN metrics, the computation of SD can be made practical and efficient on higher-resolution images by evaluating the measure on the feature-space of the images learnt by the pre-trained InceptionV3 [66] network mapping $\psi(\boldsymbol{c})$. This results in the *signed inception distance* $\text{SID}_{m,r}(\mu_p \| \mu_q) =$ given by:

$$\frac{1}{NM_{\boldsymbol{x}}} \sum_{\substack{\ell=1 \\ \boldsymbol{x}_\ell \in \mathcal{C}'_{q,r}}}^{M_{\boldsymbol{x}}} \left(\sum_{\substack{j=1 \\ \boldsymbol{c}_j \sim \mu_q}}^{N} \Phi\left(\boldsymbol{x}_\ell, \psi(\boldsymbol{c}_j)\right) - \sum_{\substack{i=1 \\ \tilde{\boldsymbol{c}}_i \sim \mu_p}}^{N} \Phi(\boldsymbol{x}_\ell, \psi(\tilde{\boldsymbol{c}}_i)) \right), \tag{3}$$

where $\mathcal{C}'_{q,r}$ denotes the hypercube of side r centered on the transformed distribution $\psi(\mu_q)$. To begin with, we find $\sigma_q = \max\{\text{diag}(\Sigma_q)\}$, where in turn, Σ_q is the covariance matrix of the samples in $\mathfrak{D}_q$. We define the hypercube $\mathcal{C}'_{q,r}$ as having side $r = \sigma_q$ along each dimension and centered around the mean of μ_q. To compare two datasets, we plot $\text{SID}_{m,r}(\mu_p \| \mu_q)$ as a function of $r \in [\sigma_q, 100\,\sigma_q]$ varying r in steps of 0.5. SID comparison figures for a few representative target datasets are given in Figure 2. We observe that, when two datasets are closely related, SID is close to zero even for small r. Datasets with lower diversity than the target have a negative SID, and vice versa. In order to quantify SID as a single number (akin to FID and KID) we consider SID, accumulated over all radii r (the cumulative SID or CSID, for short) given by: $\text{CSID}_m = \sum_r \text{SID}_{m,r}$. The last four columns of Table 1 presents CSID for $m = \lfloor \frac{n}{2} \rfloor$ for the various datasets considered. We observe that CSID is highly correlated with FID when the source is more diverse than the target, while it is able to single out sources that lack diversity, which FID cannot. These results quantitatively verify the empirical *closeness* observed when transfer-learning across datasets [53]. Additional experiments and ablation studies on SID are given in Appendices A and B.

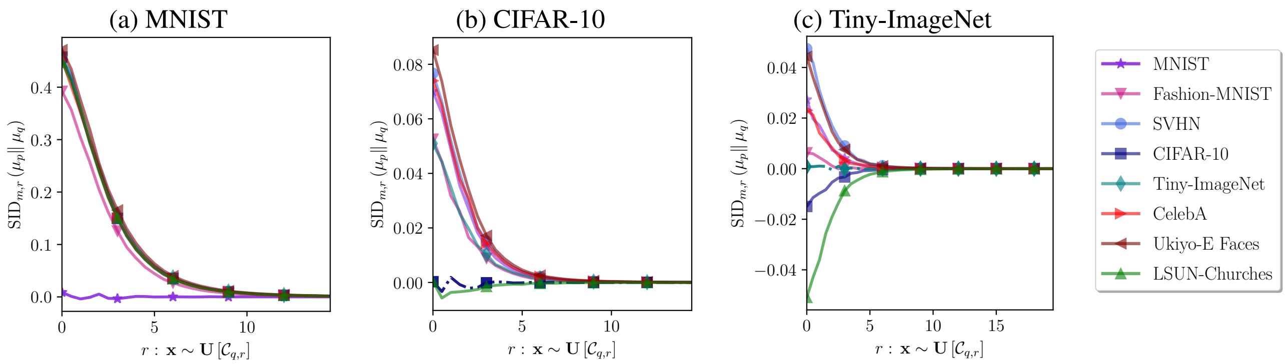

Figure 2. (Color online) $\mathrm{SID}_{m,r}$ as a function of the hyper-cube length r. We observe that Fashion-MNIST is the closest to MNIST, while Tiny-ImageNet and SVHN are closest to CIFAR-10. Fashion-MNIST and CelebA are friendly neighbors of Tiny-ImageNet.

Picking the Friendliest Neighbor: While the various approaches to compare datasets generally suggest different *friendly neighbors*, we observe that the overall trend is consistent across the measures. For example, Tiny-ImageNet and CelebA are consistently friendly neighbors to multiple datasets. We show in Sections 4 and 5 that choosing these datasets as the input indeed improves the GAN training algorithm. Both the proposed SID, and baseline FID/KID measures are relative in that they can only measure closeness between provided candidate datasets. Incorporating domain-awareness aids in the selection of appropriate input datasets between which SID can be compared. For example, all metrics identify Fashion-MNIST as a friendly neighbor when compared against color-image targets, although, as expected, the performance is sub-par in practice (cf. Section 4). One would therefore discard MNSIT and Fashion-MNIST when identifying friendly neighbors of color-image datasets. Although SID is superior to FID and KID in identifying less diverse source datasets, no single approach can always find the best dataset yet in all real-world scenarios. A pragmatic strategy is to compute various similarity measures between the target and visually/structurally similar datasets, and identify the closest one by voting.

4. Experimental Validation

To demonstrate the Spider GAN philosophy, we train *Spider* DCGAN on MNIST, CIFAR-10, and 256×256 Ukiyo-E Faces datasets using the input datasets mentioned in Section 3. While encoder-decoder architectures akin to image-to-image translation GANs could also be employed, their performance does not scale with image dimensionality. Detailed ablation experiments are provided in Appendix D.1. The second aspect is the limited stochasticity of the input dataset, when its cardinality is lower than that of the target. In these scenarios, the generator would attempt to learn one-to-many mappings between images, thereby not modeling the target entirely. For Spider DCGAN variants, the source data is resized to 16×16, vectorized, and provided as input. Based on preliminary experimentation (cf. Appendix D.2.1), to improve the input dataset diversity, we consider a Gaussian mixture centered around the samples of the source dataset formed by adding zero-mean Gaussian noise with variance $\sigma \approx 0.25$ to each source image. An alternative solution, based on pre-trained generators is presented in Section 5. We consider the Wasserstein GAN [67] loss with a one-sided gradient penalty [68]. The training parameters are described in Appendix C. In addition to FID and KID, we compare the GAN variants in terms of the cumulative SID (CSID_m) for $m = \lfloor \frac{n}{2} \rfloor$ to demonstrate the viability of evaluating GANs with the proposed SID metric.

Results: We demonstrate the ability of Spider GAN to leverage the structure present in the source dataset. From the input-output pairs given in Figure 3, we observe that, although trained in an unconstrained manner, the generator learns structurally motivated mappings. In the case when learning MNIST images with Fashion-MNIST as input, the generator has learnt to cluster similar classes, such as *Trousers* and the *1* class, or the *Shoes* class and digit *2*, which serendipitously are also visually similar. Even in scenarios where such pairwise similarity is not present, as in the case of generating Ukiyo-E Faces from CelebA or CIFAR-10, Spider GAN leverages implicit/latent structure to accelerate the generator convergence. Figure 4 presents FID as a function of iterations for each learning task for a few select target datasets. Spider GAN variants with *friendly neighborhood* inputs outperform the baseline models with parametric noise inputs, while also converging faster (up to an order in the case of MNIST). Table 2 presents the FID of the best-case models. In choosing a *friendly neighbor*, a poorly related dataset results in worse performance than the baselines, while a closely related input results in FID improvements of about 30%. The poor performance of Fashion MNIST as a friendly neighbor to CIFAR-10 and Ukiyo-E faces datasets corroborate the observations made in Section 3. We observe that CSID_m is generally in agreement with the performance indicated by FID/KID, making it a viable alternative in evaluating GANs. Experiments on remaining source-target combinations are provided in Appendix D.2.

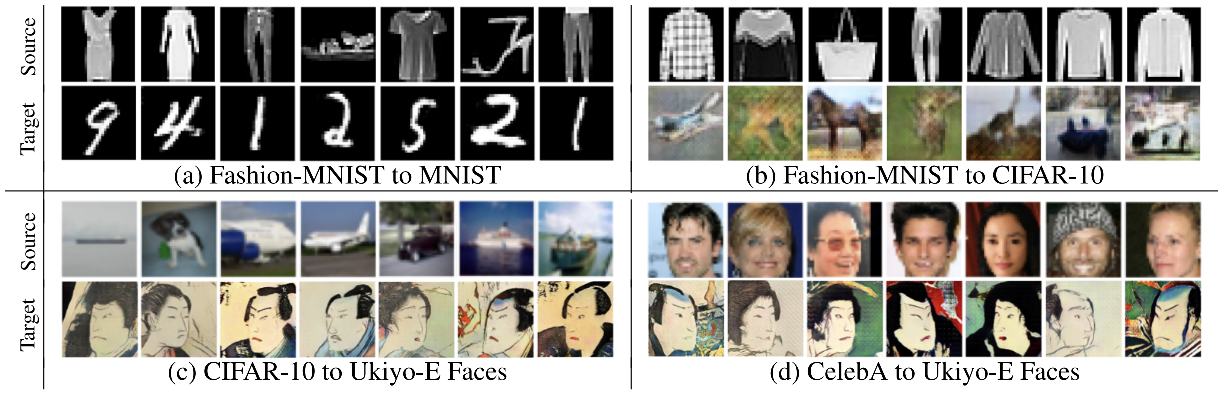

Figure 3. (Color online) Figures depicting the implicit structure learnt by Spider GAN when transforming the source to the target. The network learns both visual, and implicit correspondences across datasets. For example, the *Trouser* class in Fashion-MNIST maps to the digit *1* in MNIST, while the implicit structure is leveraged by the generator in transforming either CIFAR-10 or CelebA to Ukiyo-E Faces. A poor choice of the input distribution, for instance selecting Fashion-MNIST as the friendly neighbor of CIFAR-10, results in suboptimal learning.

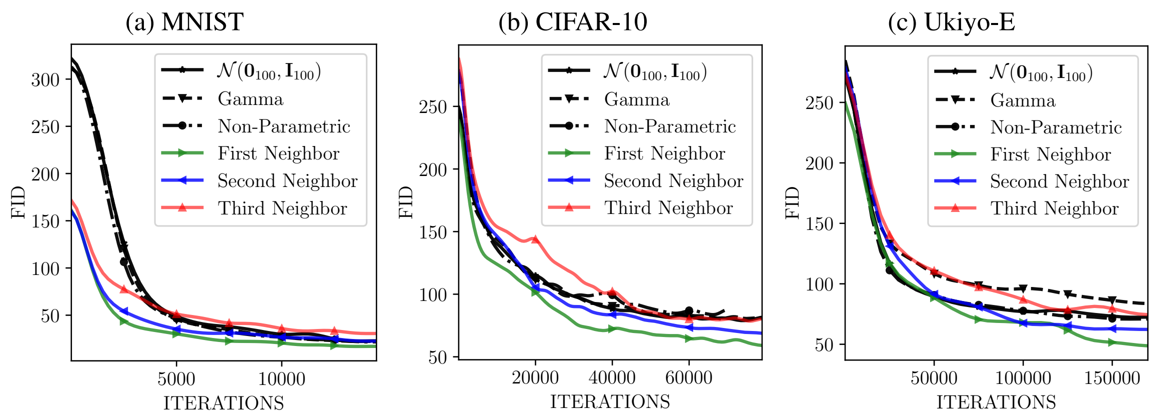

Figure 4. (Color online) FID versus iterations for training baseline and Spider GAN with the **first**, **second** and **third** friendly neighbors (color coded) identified by CSID (cf. Table 1). Using the friendliest neighbor results in the best (lowest) FID scores.On MNIST, Spider GAN variants saturate to a lower FID in an order of iterations faster than the baselines.

Table 2. Comparison of FID, KID and the proposed CSID$_m$ (with $m = \lfloor \frac{n}{2} \rfloor$) for the Spider DCGAN and baseline variants on MNIST, CIFAR-10, and Ukiyo-E Faces datasets. The first (†), second (‡) and third (⋆) *friendly neighbors* (cf. CSID; Table 1) of the target are marked for cross-referencing against the **first**, second and *third* best FID/KID/CSID$_m$ scores. Spider DCGAN, with *friendly neighborhood* input datasets outperform the baseline parametric and non-parametric priors, while a bad choice for the input results in a poorer performance.

	Input Distribution	MNIST			CIFAR10			Ukiyo-E Faces		
		FID	KID	CSID$_m$	FID	KID	CSID$_m$	FID	KID	CSID$_m$
Baselines	Gaussian [49] ($\mathbb{R}^{100}$)	21.49	0.0139	21.31	71.84	0.0619	19.90	*62.26*	0.0535	*23.10*
	Gamma [6] ($\mathbb{R}^{100}$)	21.15	0.0133	19.44	72.66	0.0483	19.87	70.02	0.0495	30.59
	Non-Parametric [9] ($\mathbb{R}^{100}$)	20.94	0.0137	20.78	74.90	0.0530	19.45	65.36	0.0421	25.40
	Gaussian ($\mathbb{R}^{H\times W\times C}$)	42.44	0.0354	32.20	73.00	0.0504	21.99	70.96	0.0501	35.30
Spider DCGAN	MNIST	–	–	–	71.70	0.0535	21.83	68.87	0.0438	33.13
	Fashion MNIST	† **16.80**	† **0.0103**	† **12.44**	77.86	0.0550	28.85	72.431	0.0455	36.21
	SVHN	27.17	0.0205	*17.23*	⋆ 64.30	⋆ 0.0451	⋆ *18.44*	70.13	0.0482	25.06
	CIFAR-10	29.22	0.0220	24.96	–	–	–	70.55	0.0530	24.12
	TinyImageNet	32.66	0.0244	36.90	† **58.82**	† **0.0305**	† **14.02**	‡ 61.91	‡ *0.0463*	‡ 21.07
	CelebA	‡ *20.55*	‡ *0.0144*	‡ 15.74	‡ 60.09	‡ *0.0434*	‡ 17.68	† **54.09**	† **0.0408**	† **20.12**
	Ukiyo-E	18.72	0.0122	19.35	67.80	0.0463	19.90	–	–	–
	LSUN-Churches	⋆ 30.67	⋆ 0.0228	⋆ 30.61	*61.46*	0.0365	19.82	⋆ 66.26	⋆ 0.0496	⋆ 25.21

Extension to Class-conditional Learning: As a proof of concept, we developed the *Spider* counterpart to the auxiliary classifier GAN (ACGAN) [50], entitled Spider ACGAN. Here, the discriminator predicts the class label of the input in addition to the *real* versus *fake* classification. We consider two variants of the generator, one without class information, and the other with the class label provided as a fully-connected embedding to the input layer. While Spider ACGAN without generator embeddings is superior to the baseline Spider GAN in learning class-level consistency, mixing between the classes is not eliminated entirely. However, with the inclusion of class embeddings in the generator, the disentanglement of classes can be achieved in Spider ACGAN. Additional details are provided in Appendix D.3. Extensions of Spider GAN to larger class-conditional GAN models such as BigGAN [69], and scenarios involving mismatch between the number of classes in the input and output datasets, are promising directions for future research.

5. Cascading Spider GANs

The DCGAN architecture employed in Section 4 does not scale well for generating high-resolution images. While training with image datasets has proven to improve the generated image quality, the improvement is accompanied by an additional memory requirement. While inference with standard GANs requires inputs drawn purely from random number generators, Spider DCGAN would require storing an additional dataset as input. To overcome this limitation, we propose a novel cascading approach, where the output distribution of a publicly available pre-trained generator is used as the input distribution to subsequent Spider GAN stages. The benefits are four-fold: First, the memory requirement is significantly lower (by an order or two), as only the weights of an input-stage generator network are required to be stored. Second, the issue of limited stochasticity in the input distribution is overcome, as infinitely many unique input samples can be drawn. Third, the network can be cascaded across architectures and styles, *i.e.,* one could employ a BigGAN input stage (trained on CIFAR-10, for example) to train a Spider StyleGAN network on ImageNet, or vice versa. *Essentially, no pre-trained GAN gets left behind*. Lastly, the cascaded Spider GANs can be coupled with existing transfer learning approaches to further improve the generator performance on small datasets [53].

5.1. Spider Variants of PGGAN and StyleGAN

We consider training the *Spider* variants of StyleGAN2 [52] and progressively growing GAN (PGGAN) [51] on small datasets, specifically the 1024-MetFaces and 1024-Ukiyo-E Faces datasets, and high-resolution FFHQ. We consider input from pre-trained GAN generators trained on the following two distributions (a) Tiny-ImageNet, based on CSID$_m$, that suggest that it is a *friendly neighbor* to the targets; and (b) AFHQ-Dogs, which possesses structural similarity to the face datasets. The experimental setup is provided in Appendix D.4, while evaluation metrics are described in Appendix C.2. To maintain consistency with the reported scores for state-of-the-art baselines models, we report only FID/KID here, and defer comparisons on CSID$_m$ to Appendix D.5. To isolate and assess the performance improvements introduced by the Spider GAN framework, we do not incorporate any augmentation or weight transfer [53]. Table 3 shows the FID values obtained by the baselines and their *Spider* variants. Spider PGGAN performs on par with the baseline StyleGAN2 in terms of FID. Spider StyleGAN2 achieves state-of-the-art FID on both Ukiyo-E and MetFaces.

To incorporate transfer learning techniques, we consider (a) learning FFHQ considering StyleGAN with adaptive discriminator augmentation (ADA) [53]; and (b) learning AFHQ-Cats considering both ADA and weight transfer [53]. Spider StyleGAN2-ADA achieves FID scores on par with the state of the art, outperforming improved sampling techniques such as Polarity-StyleGAN2 [71] and MaGNET-StyleGAN2 [72]. While StyleGAN-XL achieves marginally superior FID, it does so at the cost of a three-fold increase in network complexity [70]. The FID and KID scores, and training configurations are described in Tables 4-5. Spider StyleGAN2-ADA and Spider StyleGAN3 achieve competitive FID scores with a mere one-fifth of the training iterations. The Spider StyleGAN3 model with weight transfer achieves a state-of-the-art FID of 3.07 on AFHQ-Cats, in a fourth of the training iterations as StyleGAN3 with weight transfer. Additional results are provided in Appendix D.5.

5.2. Understanding the Spider GAN Generator

The idea of learning an optimal transformation between a pair of distributions has been explored in the context of optimal transport in *Schrödinger bridge* diffusion models [73–76]. The *closer* the two distributions are, the easier it is to learn a transport map between them. Spider GANs leverage underlying similarity, not necessarily visual, between datasets to improve generator learning. Similar discrepancies between visual features and those learnt by networks have been observed in ImageNet [77] object classification [78]. To shed more light on this intuition, consider a scenario where both the input and target datasets in Spider DCGAN are the same, with or without random noise perturbation. As expected, the generator learns an identity mapping, reproducing the input image at the output (cf. Appendix D.2.5).

Input Dataset Bias: Owing to the unpaired nature of training, Spider GANs do not enforce image-level structure to learn pairwise transformations. Therefore, the diversity of the source dataset (such as racial or gender diversity) does not affect the diversity in the learnt distribution. Experiments on Spider DCGAN with varying levels of class-imbalance in the input dataset validate this claim (cf. Appendix D.2.3).

Table 3. A comparison of the FID and KID values achieved by the PGGAN and StyleGAN2 baselines and their *Spider* variants, when trained on small datasets. A $\star$ indicates scores computed on publicly available pre-trained models using the Clean-FID library [43]. Spider StyleGAN2 achieves state-of-the-art FID and KID scores, while Spider PGGAN achieves performance comparable with the baseline StyleGAN methods.

Architecture	Input	Ukiyo-E Faces		MetFaces	
		FID	KID	FID	KID
PGGAN [51]	Gaussian	69.03	0.0762	85.74	0.0123
Spider PGGAN **Ours**)	TinyImageNet	57.63	0.0161	45.32	0.0063
StyleGAN2* [52]	Gaussian	56.74	0.0159	65.74	0.0350
StyleGAN2-ADA* [53]	Gaussian	<u>26.74</u>	<u>0.0109</u>	<u>18.75</u>	**0.0023**
Spider StyleGAN2 (**Ours**)	TinyImageNet	**20.44**	**0.0059**	**15.60**	<u>0.0026</u>
Spider StyleGAN2 (**Ours**)	AFHQ-Dogs	32.59	0.0269	29.82	0.0019

Table 4. A comparison of StyleGAN2-ADA and StyleGAN3 variants in terms of FID, on learning FFHQ. A † indicates a reported score. Spider StyleGAN2-ADA performs on par with the state-of-the-art StyleGAN-XL (three fold higher network complexity) [70], and outperforms variants with customized sampling techniques [71, 72].

Architecture	Input	FID
StyleGAN-XL [70]	Gaussian	**2.02**†
Polarity-StyleGAN2 [71]	Gaussian	2.57†
MaGNET-StyleGAN2 [72]	Gaussian	2.66†
StyleGAN2-ADA [53]	Gaussian	2.70†
Spider StyleGAN2-ADA (**Ours**)	TinyImageNet	<u>2.45</u>
Spider StyleGAN2-ADA (**Ours**)	AFHQ-Dogs	3.07
StyleGAN3-T [54]	Gaussian	2.79†
Spider StyleGAN3-T (**Ours**)	TinyImageNet	2.86

Table 5. A comparison of the FID and KID values achieved by the StyleGAN baselines and their *Spider* variants, when trained on the the AFHQ-Cats dataset, considering various training configurations. A $\star$ indicates a score reported in the Clean-FID library [43]. † Karras *et al.* only report FID on the combined AFHQv2 dataset consisting of images from the *Dogs, Cats*, and *Wild-Animals* classes. Spider StyleGAN2-ADA and Spider StyleGAN3 achieve FID and KID scores competitive with the baselines in a mere one-fifth of the training iterations, while Spider StyleGAN3 with weight transfer achieves state-of-the-art FID on AFHQ in one-fourth of the training iterations.

Architecture	Weight Transfer	Input Distribution	Training steps	FID	KID ($\times 10^{-3}$)
StyleGAN2-ADA [53]	–	Gaussian	25000	5.13*	1.54*
StyleGAN3-T [54]	–	Gaussian	25000	4.04†	–
Spider StyleGAN3-T (**Ours**)	–	AFHQ-Dogs	5000	6.29	1.64
StyleGAN2-ADA [53]	FFHQ	Gaussian	5000	3.55	<u>0.35</u>
Spider StyleGAN2-ADA (**Ours**)	FFHQ	Tiny-ImageNet	1000	3.91	1.23
StyleGAN2-ADA [53]	AFHQ-Dogs	Gaussian	5000	<u>3.47</u>*	0.37*
Spider StyleGAN2-ADA (**Ours**)	AFHQ-Dogs	Tiny-ImageNet	1500	**3.07**	**0.29**
Spider StyleGAN3-T (**Ours**)	AFHQ-Dogs	Tiny-ImageNet	1000	3.86	1.01

Input-space Interpolation: Lastly, to understand the representations learnt by Spider GANs, we consider input-space interpolation. Unlike classical GANs, where the input noise vectors are the only source of control, in cascaded Spider GANs, interpolation can be carried out at two levels. Interpolating linearly between the noise inputs to the pre-trained GAN result in a set of interpolations of the intermediate image. Transforming these images through the Spider StyleGAN generator results in greater diversity in the output images, with sharper transitions between images. This is expected as interpolating on the Gaussian manifold is known to result in discontinuities in the generated images [6, 7]. Alternatively, for fine-grained tuning, linear interpolations of the intermediate input images can be carried out, resulting in smoother transitions in the output images. Images demonstrating this behavior are provided in Appendix D.5.1. Qualitative experiments on input-space interpolation in Spider DCGAN and additional images are provided in Appendix D.2.2. These results indicate that stacking multiple Spider GAN stages yields varying levels of fineness in controlling features in the generated images.

6. Conclusions

We introduced the Spider GAN formulation, where we provide the GAN generator with an input dataset of samples from a closely related neighborhood of the target. Unlike image-translation GANs, there are no pairwise or cycle-consistency requirements in Spider GAN, and the trained generator learns a transformation from the underlying latent data distribution to the target data. While the *best* input dataset is a problem-specific design choice, we proposed approaches to identify promising friendly neighbors. We proposed a novel signed inception distance, which measures the relative diversity between two datasets. Experimental validation showed that Spider GANs, trained with closely related datasets, outperform baseline GANs with parametric input distributions, achieving state-of-the-art FID on Ukiyo-E Faces, MetFaces, FFHQ and AFHQ-Cats.

While we focused on adaptive augmentation and weight transfer, incorporating other transfer learning approaches [29, 60, 79] is a promising direction for future research. One could also explore extensions to vector quantized GANs [80, 81] or high-resolution class-conditional GANs [69, 82].

References

[1] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. C. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in Neural Information Processing Systems 27*, pp. 2672–2680, 2014.

[2] L. Tran, X. Yin, and X. Liu, "Disentangled representation learning GAN for pose-invariant face recognition," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1283–1292, 2017.

[3] E. Agustsson, A. Sage, R. Timofte, and L. V. Gool, "Optimal transport maps for distribution preserving operations on latent spaces of generative models," in *Proceedings of the 7th International Conference on Learning Representations*, 2019.

[4] S. Gurumurthy, R. K. Sarvadevabhatla, and R. V. Babu, "DeLiGAN: Generative adversarial networks for diverse and limited data," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, July 2017.

[5] T. White, "Sampling generative networks," *arXiv preprints, arXiv:1609.04468*, 2016.

[6] Y. Kilcher, A. Lucchi, and T. Hofmann, "Semantic interpolation in implicit models," in *Proceedings of the 6th International Conference on Learning Representations*, 2018.

[7] D. Leśniak, I. Sieradzki, and I. Podolak, "Distribution-interpolation trade off in generative models," in *Proceedings of the 7th International Conference on Learning Representations*, 2019.

[8] M. Kuznetsov, D. Polykovskiy, D. P. Vetrov, and A. Zhebrak, "A prior of a googol gaussians: a tensor ring induced prior for generative models," in *Advances in Neural Information Processing Systems 32*, 2019.

[9] R. Singh, P. Turaga, S. Jayasuriya, R. Garg, and M. Braun, "Non-parametric priors for generative adversarial networks," in *Proceedings of the 36th International Conference on Machine Learning*, vol. 97, pp. 5838–5847, June 2019.

[10] A. B. L. Larsen, S. K. Søonderby, H. Larochelle, and O. Winther, "Autoencoding beyond pixels using a learned similarity metric," in *Proceedings of The 33rd International Conference on Machine Learning*, vol. 48, Jun 2016.

[11] G. Parmar, D. Li, K. Lee, and Z. Tu, "Dual contradistinctive generative autoencoder," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2021.

[12] M. Khayatkhoei, M. K. Singh, and A. Elgammal, "Disconnected manifold learning for generative adversarial networks," in *Advances in Neural Information Processing Systems 31*, pp. 7343–7353, 2018.

[13] X. Chen, Y. Duan, R. Houthooft, J. Schulman, I. Sutskever, and P. Abbeel, "InfoGAN: Interpretable representation learning by information maximizing generative adversarial nets," in *Advances in Neural Information Processing Systems 29*, pp. 2180–2188, 2016.

[14] T. Karras, S. Laine, and T. Aila, "A style-based generator architecture for generative adversarial networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, June 2019.

[15] Y. Shen, J. Gu, X. Tang, and B. Zhou, "Interpreting the latent space of GANs for semantic face editing," in *Proceeding of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9240–9249, 2020.

[16] J.-Y. Zhu, R. Zhang, D. Pathak, T. Darrell, A. A. Efros, O. Wang, and E. Shechtman, "Toward multimodal image-to-image translation," in *Advances in Neural Information Processing Systems 30*, pp. 465–476, 2017.

[17] K. Bousmalis, N. Silberman, D. Dohan, D. Erhan, and D. Krishnan, "Unsupervised pixel-level domain adaptation with generative adversarial networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, July 2017.

[18] E. Tzeng, J. Hoffman, K. Saenko, and T. Darrell, "Adversarial discriminative domain adaptation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2962–2971, 2017.

[19] Z. Murez, S. Kolouri, D. Kriegman, R. Ramamoorthi, and K. Kim, "Image to image translation for domain adaptation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4500–4509, 2018.

[20] B. Khurana, S. R. Dash, A. Bhatia, A. Mahapatra, H. Singh, and K. Kulkarni, "SemIE: Semantically-aware image extrapolation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 14900–14909, October 2021.

[21] S. Hicsönmez, N. Samet, E. Akbas, and P. Duygulu, "GANILLA: Generative adversarial networks for image to illustration translation," *Image and Vision Computing*, vol. 95, p. 103886, Feb, 2020.

[22] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," *arXiv preprints, arXiv:1611.07004*, 2018.

[23] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *Proceedings of International Conference on Computer Vision*, 2017.

[24] M.-Y. Liu, T. Breuel, and J. Kautz, "Unsupervised image-to-image translation networks," in *Advances in Neural Information Processing Systems 30*, pp. 700–708, 2017.

[25] Z. Yi, H. Zhang, P. Tan, and M. Gong, "DualGAN: Unsupervised dual learning for image-to-image translation," in *Proceedings of the International Conference on Computer Vision*, Oct. 2017.

[26] J. Hoffman, E. Tzeng, T. Park, J.-Y. Zhu, P. Isola, K. Saenko, A. A. Efros, and T. Darrell, "CyCADA: Cycle-consistent adversarial domain adaptation," in *Proceedings of the 35th International Conference on Machine Learning*, vol. 80, pp. 1989–1998, July 2018.

[27] H.-Y. Lee, H.-Y. Tseng, J.-B. Huang, M. Singh, and M.-H. Yang, "Diverse image-to-image translation via disentangled representations," in *Proceedings of the European Conference on Computer Vision*, 2018.

[28] X. Huang, M.-Y. Liu, S. Belongie, and J. Kautz, "Multimodal unsupervised image-to-image translation," in *Proceedings of the European Conference on Computer Vision*, Sep. 2018.

[29] U. Ojha, Y. Li, C. Lu, A. A. Efros, Y. J. Lee, E. Shechtman, and R. Zhang, "Few-shot image generation via cross-domain correspondence," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[30] Y. Choi, Y. Uh, J. Yoo, and J.-W. Ha, "Stargan v2: Diverse image synthesis for multiple domains," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[31] C. Fefferman, S. Mitter, and H. Narayanan, "Testing the manifold hypothesis," *Journal of the American Mathematical Society*, vol. 29, pp. 983–1049, 2016.

[32] J. Liang, J. Yang, H.-Y. Lee, K. Wang, and M.-H. Yang, "SubGAN: An unsupervised generative model via subspaces," in *Proceedings of the European Conference on Computer Vision*, September 2018.

[33] U. Tanielian, T. Issenhuth, E. Dohmatob, and J. Mary, "Learning disconnected manifolds: a no GANs land," *arXiv preprints, arXiv:2006.04596*, 2020.

[34] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, and A. Y. Ng, "Reading digits in natural images with unsupervised feature learning," in *NIPS Workshop on Deep Learning and Unsupervised Feature Learning*, 2011.

[35] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[36] J. A. Thomas and T. M. Cover, *Elements of Information Theory*. John Wiley and Sons, Ltd, 2005.

[37] J. L. Kelley, *General Topology*. Courier Dover Publications, Inc., 2017.

[38] H. Xiao, K. Rasul, and R. Vollgraf, "Fashion-MNIST: A novel image dataset for benchmarking machine learning algorithms," *arXiv preprint, arXiv:1708.07747*, Aug. 2017.

[39] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, "GANs trained by a two time-scale update rule converge to a local Nash equilibrium," *arXiv preprints, arXiv:1706.08500*, 2018.

[40] M. Bińkowski, D. J. Sutherland, M. Arbel, and A. Gretton, "Demystifying MMD GANs," in *Proceedings of the 6th International Conference on Learning Representations*, 2018.

[41] T. Unterthiner, B. Nessler, C. Seward, G. Klambauer, M. Heusel, H. Ramsauer, and S. Hochreiter, "Coulomb GANs: Provably optimal Nash equilibria via potential fields," in *Proceedings of the 6th International Conference on Learning Representations*, 2018.

[42] S. Asokan and C. S. Seelamantula, "LSGANs with gradient regularizers are smooth high-dimensional interpolators," in *Proceedings of the "First Workshop on Interpolation and Beyond" at NeurIPS*, 2022.

[43] G. Parmar, R. Zhang, and J.-Y. Zhu, "On buggy resizing libraries and surprising subtleties in FID calculation," *arXiv preprint, arXiv:2104.11222*, vol. abs/2104.11222, April 2021.

[44] A. Krizhevsky, "Learning multiple layers of features from tiny images," *Master's thesis, University of Toronto*, 2009.

[45] Y. Le and X. Yang, "Tiny imagenet visual recognition challenge," 2015.

[46] F. Yu, A. Seff, Y. Zhang, S. Song, T. Funkhouser, and J. Xiao, "LSUN: Construction of a large-scale image dataset using deep learning with humans in the loop," *arXiv preprints, arXiv:1506.03365*, 2016.

[47] Z. Liu, P. Luo, X. Wang, and X. Tang, "Deep learning face attributes in the wild," in *Proceedings of International Conference on Computer Vision*, 2015.

[48] J. N. M. Pinkney and D. Adler, "Resolution dependent GAN interpolation for controllable image synthesis between domains," *arXiv preprint, arXiv:2010.05334*, Oct. 2020.

[49] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," in *Proceedings of the 4th International Conference on Learning Representations*, 2016.

[50] A. Odena, C. Olah, and J. Shlens, "Conditional image synthesis with auxiliary classifier GANs," in *Proceedings of the 34th International Conference on Machine Learning (ICML)*, 2017.

[51] T. Karras, T. Aila, S. Laine, and J. Lehtinen, "Progressive growing of GANs for improved quality, stability, and variation," in *Proceedings of the 6th International Conference on Learning Representations*, 2018.

[52] T. Karras, S. Laine, M. Aittala, J. Hellsten, J. Lehtinen, and T. Aila, "Analyzing and improving the image quality of StyleGAN," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[53] T. Karras, M. Aittala, J. Hellsten, S. Laine, J. Lehtinen, and T. Aila, "Training generative adversarial networks with limited data," in *Advances in Neural Information Processing Systems 33*, 2020.

[54] T. Karras, M. Aittala, S. Laine, E. Härkönen, J. Hellsten, J. Lehtinen, and T. Aila, "Alias-free generative adversarial networks," in *Advances in Neural Information Processing Systems*, June 2021.

[55] A. Blum, J. Hopcroft, and R. Kannan, *Foundations of Data Science*. Cambridge University Press, 2013.

[56] M. Hein and J.-Y. Audibert, "Intrinsic dimensionality estimation of submanifolds in R^d," in *Proceedings of the 22nd International Conference on Machine Learning*, pp. 289–296, 2005.

[57] R. Feng, Z. Lin, J. Zhu, D. Zhao, J. Zhou, and Z.-J. Zha, "Uncertainty principles of encoding GANs," in *Proceedings of the 38th International Conference on Machine Learning*, pp. 3240–3251, Jul 2021.

[58] E. Facco, M. d'Errico, A. Rodriguez, and A. Laio, "Estimating the intrinsic dimension of datasets by a minimal neighborhood information," *Scientific Reports*, vol. 7, Sep. 2017.

[59] A. Casanova, M. Careil, J. Verbeek, M. Drozdzal, and A. R. Soriano, "Instance-conditioned GAN," in *Advances in Neural Information Processing Systems 34*, pp. 27517–27529, 2021.

[60] Y. Wang, A. Gonzalez-Garcia, D. Berga, L. Herranz, F. S. Khan, and J. Weijer, "MineGAN: Effective knowledge transfer from GANs to target domains with few images," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, June 2020.

[61] F. Camastra and A. Staiano, "Intrinsic dimension estimation: Advances and open problems," *Information Sciences*, vol. 328, pp. 26–41, 2016.

[62] Y. Wang, C. Wu, L. Herranz, J. van de Weijer, A. Gonzalez-Garcia, and B. Raducanu, "Transferring GANs: Generating images from limited data," in *Proceedings of the European Conference on Computer Vision*, pp. 220–236, 2018.

[63] M. S. M. Sajjadi, O. Bachem, M. Lucic, O. Bousquet, and S. Gelly, "Assessing generative models via precision and recall," in *Advances in Neural Information Processing Systems 31*, pp. 5228–5237, 2018.

[64] T. Kynkäänniemi, T. Karras, S. Laine, J. Lehtinen, and T. Aila, "Improved precision and recall metric for assessing generative models," in *Advances in Neural Information Processing Systems 32*, 2019.

[65] N. Aronszajn, T. Creese, and L. Lipkin, *Polyharmonic Functions*. Oxford: Clarendon, 1983.

[66] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," *arXiv preprint, arXiv:1512.00567*, Dec. 2015.

[67] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein generative adversarial networks," in *Proceedings of the 34th International Conference on Machine Learning*, pp. 214–223, 2017.

[68] L. Mescheder, A. Geiger, and S. Nowozin, "Which training methods for GANs do actually converge?," in *Proceedings of the 35th International Conference on Machine Learning*, vol. 80, pp. 3481–3490, 2018.

[69] A. Brock, J. Donahue, and K. Simonyan, "Large scale GAN training for high fidelity natural image synthesis," *arXiv preprints, arXiv:1809.11096*, Sep. 2018.

[70] A. Sauer, K. Schwarz, and A. Geiger, "StyleGAN-XL: Scaling StyleGAN to large diverse datasets," *arXiv.org*, vol. abs/2201.00273, 2022.

[71] A. I. Humayun, R. Balestriero, and R. Baraniuk, "Polarity sampling: Quality and diversity control of pre-trained generative networks via singular values," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2022.

[72] A. I. Humayun, R. Balestriero, and R. Baraniuk, "MaGNET: Uniform sampling from deep generative network manifolds without retraining," in *International Conference on Learning Representations (ICLR)*, 2022.

[73] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," *preprint arxiv:2006.11239*, 2020.

[74] F. Vargas, P. Thodoroff, A. Lamacraft, and N. Lawrence, "Solving Schrödinger bridges via maximum likelihood," *Entropy*, vol. 23, 2021.

[75] V. D. Bortoli, J. Thornton, J. Heng, and A. Doucet, "Diffusion Schrödinger bridge with applications to score-based generative modeling," in *Advances in Neural Information Processing Systems*, 2021.

[76] T. Chen, G.-H. Liu, and E. Theodorou, "Likelihood training of Schrödinger bridge using forward-backward SDEs theory," in *International Conference on Learning Representations*, 2022.

[77] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2009.

[78] T. Fel, I. Felipe, D. Linsley, and T. Serre, "Harmonizing the object recognition strategies of deep neural networks with humans," *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

[79] Y. Li, R. Zhang, J. Lu, and E. Shechtman, "Few-shot image generation with elastic weight consolidation," in *Advances in Neural Information Processing Systems*, 2020.

[80] P. Esser, R. Rombach, and B. Ommer, "Taming transformers for high-resolution image synthesis," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2021.

[81] J. Yu, X. Li, J. Y. Koh, H. Zhang, R. Pang, J. Qin, A. Ku, Y. Xu, J. Baldridge, and Y. Wu, "Vector-quantized image modeling with improved VQGAN," in *Proceedings of the 10th International Conference on Learning Representations*, 2022.

[82] M. Kang, W. Shim, M. Cho, and J. Park, "Rebooting ACGAN: Auxiliary classifier GANs with stable training," in *Advances in Neural Information Processing Systems 34*, 2021.

[83] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proceedings of the 3rd International Conference on Learning Representations*, 2015.

[84] M. Abadi *et al.*, "TensorFlow: Large-scale machine learning on heterogeneous distributed systems," *arXiv preprint, arXiv:1603.04467*, Mar. 2016.

[85] T. Liang, "How well generative adversarial networks learn distributions," *Journal of Machine Learning Research*, vol. 22, no. 228, pp. 1–41, 2021.

[86] N. Schreuder, V.-E. Brunel, and A. Dalalyan, "Statistical guarantees for generative models without domination," in *Proceedings of the 32nd International Conference on Algorithmic Learning Theory*, vol. 132, pp. 1051–1071, Mar 2021.