



# **Survey on Implementations of Generative Adversarial Networks for Semi-Supervised Learning**

Ali Reza Sajun \*២ and Imran Zualkernan ២

Computer Science and Engineering Department, American University of Sharjah, Sharjah P.O. Box 26666, United Arab Emirates; izualkernan@aus.edu

\* Correspondence: b00068908@aus.edu

Abstract: Given recent advances in deep learning, semi-supervised techniques have seen a rise in interest. Generative adversarial networks (GANs) represent one recent approach to semi-supervised learning (SSL). This paper presents a survey method using GANs for SSL. Previous work in applying GANs to SSL are classified into pseudo-labeling/classification, encoder-based, TripleGAN-based, two GAN, manifold regularization, and stacked discriminator approaches. A quantitative and qualitative analysis of the various approaches is presented. The R3-CGAN architecture is identified as the GAN architecture with state-of-the-art results. Given the recent success of non-GAN-based approaches for SSL, future research opportunities involving the adaptation of elements of SSL into GAN-based implementations are also identified.

Keywords: generative adversarial networks; semi-supervised learning; deep learning



Citation: Sajun, A.R.; Zualkernan, I. Survey on Implementations of Generative Adversarial Networks for Semi-Supervised Learning. *Appl. Sci.* 2022, *12*, 1718. https://doi.org/ 10.3390/app12031718

Academic Editors: Andrea Prati and Mauro Castelli

Received: 20 November 2021 Accepted: 28 January 2022 Published: 7 February 2022

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https:// creativecommons.org/licenses/by/ 4.0/).

# 1. Introduction

With recent advances in deep learning and its applications, research opportunities in the area have expanded and diversified in different directions. One of these directions is semi-supervised learning (SSL). As opposed to supervised learning, SSL is a form of learning that can learn based on incomplete data where only some of the data is labelled [1]. In supervised learning, the training data consist of a set of data points and a corresponding label for each of the points. Conversely, in unsupervised learning, the training data consist of only data points with no output provided, therefore requiring a process that discovers unknown structures and groupings within the data [2]. Semi-supervised learning is used in situations where there are a small number of labeled training samples along with a large number of unlabeled data points available [3]. While supervised learning has been the dominant technique used for most classification tasks, labeled data can often be difficult to obtain, and the process of labeling data can be very expensive and time consuming [4]. Therefore, SSL obviates the need for large, labelled datasets by using some labelled but mostly unlabeled data.

Semi-supervised learning relies on the assumption that the data distribution over the input space embeds significant information about the distribution of the labels in the output space [1]. Most SSL algorithms will break down if this assumption is not met as the input space would not contain any information about the actual labels and, therefore, improving accuracy with the help of unlabeled data would not be possible.

As [4] reports, if the sample distribution of the data does not embed significant information, then the resulting learning might not show improvement when compared to supervised learning, and may lead to an increase in false predictions. The basic assumption can further be sub-divided into three assumptions; the smoothness assumption, low-density assumption, and manifold assumption.

The first assumption, called the smoothness assumption, states that given two data points that are close by in the input space, the corresponding labels in the output space

should be the same for both points [1]. This assumption is sometimes referred to as the cluster assumption, which states that data points of each class must form a cluster where points can be connected by short curves that do not pass through low-density regions [4]. Consequently, on the basis of this assumption, the decision boundary should not cross high density areas but rather lie in low-density regions, which is also the basis for the low-density assumption discussed later [5]. This can be visualized in Figure 1 where, given the green cross points and yellow triangle points are each in their respective clusters, the assumption would be that the label is also the same, in addition to the decision boundary lying in the low-density area between the two clusters.



Figure 1. Smoothness and Low-Density Assumption.

The second assumption, called the low-density assumption, states that the decision boundary in a classifier should pass through low-density regions in the input space [1]. This is also related to the cluster assumption since if the decision boundary is to pass through areas of high density, it would cut the cluster into different classes and therefore violate the cluster assumption [4]. Additionally, the low-density assumption is consistent with the smoothness assumption, which can be demonstrated by assuming a low-density area in the input space where the probability of a data point existing is low. This assumption can be visualized in Figure 1 where the optimal decision boundary is shown to be in the low-density area in between the two well-defined clusters.

The points in a high dimensional space can be mapped to low-dimensional structures known as manifolds. For example, a 3-dimensional input space where all data points lie on a sphere can be mapped to a 2-dimensional manifold [1]. The manifold assumption states that the input space of the data consists of multiple manifolds of low dimensions on which all data points lie. Furthermore, it states that any data points lying on the same manifold belong to the same class [4]. Therefore, if the manifolds are determined, and the unlabeled data points are distributed on these manifolds, the class labels can be inferred based on which manifold an unlabeled data point lies.

A multitude of SSL algorithms based on these three assumptions have been proposed yielding excellent results on datasets commonly used for benchmarking such as CIFAR [6] and SVHN [7], with recent algorithms such as FixMatch [8] showing error rates as low as 4.26% for CIFAR-10 and 2.28% for SVHN.

Generative adversarial networks (GANs) represent another class of techniques employed for SSL. The next section discusses common SSL techniques and the viability of generative architectures in the semi-supervised learning scenarios.

# 2. Common Techniques Used in Semi-Supervised Learning

A number of algorithms and approaches to semi-supervised learning have been proposed recently. These algorithms can be grouped into different classes depending on criteria like the assumptions they are based on, the way they make use of unlabeled data, and how they relate to supervised algorithms [1]. However, most algorithms use a common set of techniques including consistency regularization, pseudo-labeling, and entropy minimization. These techniques are briefly described below.

## 2.1. Consistency Regularization

Consistency regularization [4] is an important technique used in SSL and relies on the manifold and smoothness assumptions. The technique assumes that realistic perturbations of data points in the input space (vi data augmentation, for example) should not significantly change the predicted labels of the model [9]. In simpler terms, if an input is disturbed in a way that preserves its semantics using operations such as image flipping or cropping, for example, the output label should be close to the output label for the original image. The idea is operationalized by adding a consistency regularization term to the loss function [4] that penalizes any sensitivity the model shows to the various perturbations [10].

The initial implementation of consistency regularization for deep SSL is most commonly attributed to Sajjadi et al. [11] where random augmentations were applied to the same data sample that forced predictions to be similar by proposing an unsupervised loss function that minimized the mean-squared difference between different passes of a single data point through the network. Additionally, another loss function called the mutual-exclusivity loss was used to ensure that the model's prediction vector had only one non-zero element, thereby forcing each prediction to be valid and non-ambiguous. Subsequently, the idea of temporal ensembling was introduced by Lain et al. [12], which used an exponential moving average of historical predictions at different epochs of training as one part of the output. However, the downside of this method was that predictions would change only after an entire epoch, which was troublesome in the case of large datasets. Therefore, the mean teacher model was proposed, which averaged model weights instead of previous predictions [13]. An alternate approach was proposed by Lou et al. [14] who proposed adding an additional regularization in the form of a contrastive loss on the predictions, thus forcing predictions to be different when the data points were from different classes. Another interesting approach was proposed by Miyato et al. [15] in which virtual adversarial training (VAT) was used to add perturbations to the data in order to achieve consistency regularization on the model predictions. An adversarial dropout was introduced by Perk et al. [16] that involved a dropout mask being learnt for data perturbation in a direction adversarial to the model's virtual label assignment. More recent work includes Verma et al. [17] proposing interpolation consistency training that encourages predictions at the interpolated data sample pairs to be consistent with the interpolated predictions, which helps move the decision boundary to low-density regions of the data space. A recent approach involving the use of consistency regularization was proposed in ReMixMatch [18], which did so by strongly augmenting an input multiple times and training the model to encourage the prediction for all strongly augmented images to be consistent with the prediction for a weakly augmented version of the same image.

Given the importance of the aforementioned techniques in the area of semi-supervised learning, numerous GAN-based SSL approaches have also leveraged these techniques. Consistency regularization was used by a number of GAN-based solutions such as Wei et al. [19] that added a consistency term to the loss function inspired by temporal ensembling [12]. Similarly, Chen et al. [20] reported that GAN-based SSL techniques lagged behind other SSL techniques due to a lack of consistency in class probability predictions for the same image under local perturbations. The authors attempted to solve the issue by adding an auxiliary loss term to the discriminator, which accounted for consistency regularization by using an approach based on the Mean Teacher [13]. Zhang et al. [10] proposed CR-GAN by adding consistency regularization to the discriminator while training by randomly

augmenting training images as they were passed to the discriminator, and penalizing the sensitivity of the discriminator to the augmentations. Zhao et al. [21] argued that this approach was flawed as the consistency was applied only to real images and not to generated images, which could result in the generator learning the augmentation features, and introducing them into the generated images. They proposed an improved consistency regularization technique that added a consistency term to the discriminator for both real and generated images. Furthermore, they proposed an additional level of consistency by encouraging the generator to be sensitive to augmented latent vectors while encouraging the discriminator to be insensitive. Therefore, with the recent work adding consistency regularization to GANs, it can serve only to improve the adaptability of the technique towards semi-supervised learning.

### 2.2. Pseudo-Labeling

Pseudo-labeling [22] is a simple technique involving training the model on the labeled data and using the model to make predictions for the unlabeled data. The model predictions are then used as labels for the unlabeled data for further supervised training. Pseudo-labels are produced by setting a predefined threshold for assigning a class to an unlabeled sample, which can then be used as targets for a standard supervised loss function [9].

While this is the simplest technique theoretically, a number of attempts have been made to adapt this approach as part of a more evolved algorithm towards SSL. For example, Shi et al. [23] used class predictions as hard labels for the unlabeled data in addition to introducing an uncertainty weight for each sample loss. A more recent approach Iscen et al. [24] employed a graph-based transductive label propagation method on the basis of the manifold assumption to make predictions on the entire data, and then use these predictions as pseudo-labels. This technique was also used in the FixMatch algorithm [8] that generated pseudo-labels by passing weakly augmented unlabeled data through the model and using the predictions as labels when training strongly augmented versions of the same samples. A slightly different approach was proposed byArazo et al. [25] that proposed using soft pseudo-labels using the network's latest predictions.

Pseudo-labeling has been used in GANs performing SSL. For example, one such implementation was TripleGAN [26] where pseudo-labels were generated for unlabeled data and used as a real sample for the discriminator. This was carried out to prevent the discriminator from memorizing the empirical distribution of the labeled data. Similarly, Dong et al. [27] implemented pseudo-labeling for both unlabeled and generated images, which was then used along with cross-entropy during the training process. Finally, Liu et al. [28] used pseudo-labeling as well as part of the R<sup>3</sup>-CGAN model; pseudo-labeling was used to assign labels to the unlabeled classes.

### 2.3. Entropy Minimization

Entropy minimization is the process by which the network is encouraged to make high confidence predictions on the unlabeled data regardless of the predicted class [3]. This technique discourages the decision boundary from passing near data points as a line passing near data points would produce low confidence predictions [9]. This idea is operationalized by adding a loss term that minimizes the entropy of the prediction function. While entropy minimization ideally discourages the decision boundary from passing close to data points, Oliver et al. [9] reported an issue seen in high capacity models such as neural networks where the decision boundary overfits to locally avoid a number of small data points. Therefore, Ouali et al. [3] suggested that on its own entropy minimization was not as effective in producing viable results. However, this technique could be used in combination with other semi-supervised learning techniques as part of an algorithm to produce state-of-the-art results.

The implementation of entropy minimization with GANs performing SSL has also been seen in the literature, albeit less commonly. One notable implementation was Dai et al. [29] where the authors reported adding a conditional entropy term to the discriminator's objective function in order to strengthen the discriminator's true/fake belief following the approach of virtual adversarial training [15].

### 3. Literature Review of GANS for SSL

# 3.1. Taxonomy

Figure 2 shows a taxonomy of the surveyed papers. As Figure 2 shows, early approaches to SSL GANs generally involved extensions to existing GAN models by use of pseudo-labeling, or by adding a classifier component to the original GAN architecture. This approach was seen in numerous models such as CatGAN [30], SGAN [31], Improved GAN [32], GoodBadGAN [29], CT-GAN [19], and MatchGAN [33]. Many others used a conditional approach where the image as well as the label was fed into the GAN. This was seen in the case of EnhancedTGAN [34], MarginGAN [27], Triangle-GAN [35], Structured GAN [36], R<sup>3</sup>-CGAN [28], and EC-GAN [37]. A third approach consisted of models using encoder-based approaches where an encoder was added to the GAN architecture to map images into a latent space, which then subsequently helped in the training process. This approach was seen in BiGAN [38], ALI [39], and Augmented BiGAN [40] models. More recent approaches have used manifold regularization techniques in order to make the model more resistant to perturbations in the input. Laplacian-based GAN [41], Monte Carlo-based GAN [42], SelfAttentionGAN [43], and SSVM-GAN [44] all fall into this category. Other unique approaches involved using two GANs as seen in MCGAN [45], VTGAN [46], and IAGAN [47], and finally leveraging conditional GANs in a stacked discriminator approach, seen in SS-GAN [48], to discriminate between predicted attributes.



Figure 2. Taxonomy of surveyed papers.

### 3.2. Notation

The notation and symbols used within this paper are defined in Table 1.

Term	Definition	
X	Original labeled data points	
у	Original labels	
x <sub>u</sub>	Unlabeled data points	
y'	Labels of generated data	
Z	Randomly generated latent space	
G(z)	Generator	
E(z)	Encoder	
D	Discriminator	
С	Classifier	
P(y)	Probability—Discriminator output	
P(c)	Probability—Classifier output	
H(x)	Entropy of a given distribution over data x	
$E_{x \sim p_{data(x)}}$	Expected value given x distributed as $p_{data(x)}$	
$\parallel f \parallel^2_L$	Laplacian Norm	

Table 1. Notations used.

### 3.3. Extensions Using Pseudo-Labeling and Classifiers

GANs were introduced by Goodfellow et al. [49] as an architecture involving a generator and a discriminator competing against each other, with the generator generating fake images, and the discriminator identifying them as fake. As Engelen et al. [1] notes, GANs are good candidates for SSL because the generator is trained on unlabeled images and the discriminator's primary function is to assess the quality of the generator. While the original implementations focused on using the GAN framework for image generation, it was not long before CatGAN [30] was proposed in 2015 that added an unsupervised classifier to the proposed model in order to enable categorical classification using a cross-entropy loss. This paper also added a cross-entropy loss term for the labeled samples that penalized misclassifications of real data. This approach was also used in SGAN [31], that leveraged a single discriminator/classifier network by having N + 1 classifying neurons, where N is the number of classes and one neuron is added to identify fake samples.

Improved GAN [32] introduced feature matching, which involved training the generator to produce images that match the expected value of features at an intermediate layer of the discriminator instead of for the final layer. This approach prevented the generator from overtraining to the specific discriminator. Mini-batch discrimination was also proposed where the discriminator predicted whether a mini-batch of images were real or fake instead of individually evaluating single images. This helped in making the generator produce more varied samples since the generator raced to the one point that the discriminator believed was realistic. Mini-batch discrimination generated better images. However, feature matching worked much better for the SSL component. In addition to the proposed techniques, the authors also argued that training GANs using gradient descent techniques was counterintuitive as they were designed to minimize the cost function instead of finding the Nash equilibrium. This argument is an important precursor to subsequent work that tried to reach a balance between generators and discriminators. For example, GoodBadGAN [29] was based on the premise that obtaining good classifier performance, and an effective generator at the same time was difficult, and therefore the focus should be on achieving one outcome only. They based their argument on Salmins et al. [32] and noted that while mini-batch discrimination produced better images, it was feature matching that showed an improved performance for SSL. They also questioned training the discriminator and generator jointly, and demonstrated that a good discriminator could be produced by using a bad generator. This was first carried out by increasing the generator entropy by adding

an auxiliary cost in addition to forcing the generator to produce samples closer to the decision boundary, which was achieved by adding a term to the generator's objective function that penalized high density samples. This pushed the generated samples to move towards low-density areas. The final generator objective function was defined as shown in Equation (1).

$$\min_{C} - H(p_{G}) + \mathbb{E}_{x \sim p_{G}} \log_{p}(x) \mathbf{I}[p(x) > \epsilon] + \| \mathbb{E}_{x \sim p_{G}} f(x) - \mathbb{E}_{x \sim u} f(x) \|^{2}$$
(1)

Another set of initial studies used Wasserstein GANs [50] as a baseline model for SSL. For example, CT-GAN [19] used a Wasserstein distance function, which seems to work better for learning distributions supported by low-dimensional manifolds as opposed to contemporary functions such as the Jensen–Shannon divergence used by many GANs. The Wasserstein distance converts the discriminator to a real-valued set of 1-Lipschitz functions instead of being a classifier. Wasserstein distance was used in conjunction with consistency regularization by Lane et al. [12]. They used a discriminator similar to that of Salimans et al. [32] with an output size of K + 1 neurons where K was the number of classes. Additionally, a consistency term was added to the loss function that forced consistency between multiple augmentations of the same data point. The objective function for the discriminator can be seen in Equation (2).

$$L_{semi_{dis}} = -E_{x,y \sim P_{x,y}}[\log D(y|x)] - E_{z \sim P_{z}}[\log D(K+1|G(z))] - E_{x \sim P_{r}}[\log(1 - D(K+1|x))] + \lambda CT|_{x',x''}$$
(2)

MatchGAN [33] also used Wasserstein distance and was a semi-supervised conditional GAN that made use of the label space in the target domain in conjunction with unlabeled samples to generate additional labeled samples. They reported using a system in which labels from the pool of labeled samples were assigned to unlabeled samples and passed through the generator that created synthetic versions of the images on the basis of the target labels. A match loss term was added, which compared the generated images to the original labeled image from which the target label was sampled.

# 3.4. Encoder-Based Approaches

The encoder-based approach was first presented as part of BiGAN [38], where the authors argued that while GANs were effective at taking a latent space and generating data, there was no technique for GANs to project the data back into the latent space. Therefore, they proposed an approach where an encoder was included as part of the GAN architecture to generate a latent space mapping from the input data. The architecture of the BiGAN model is shown in Figure 3.



Figure 3. The BiGAN Architecture.

The adopted approach involved the discriminator receiving a pair of latent space mapping and data as input wherein it discriminated jointly the data and latent space with the latent component either being the generator input z or the encoder output E(x). The training objective for this architecture is shown in Equation (3).

$$\min_{G,E} \max_{D} E_{x \sim p_{x}} \underbrace{\left[ E_{z \sim p_{E}(\cdot|x)} [\log D(x,z)] \right]}_{\log D(x,E(x))} + E_{z \sim p_{z}} \underbrace{\left[ E_{x \sim p_{G}(\cdot|z)} [\log(1 - D(x,z))] \right]}_{\log (1 - D(G(z),z))}$$
(3)

Adversarially learned inference (ALI) [39] also used an encoder that authors referred to as an "inference machine" that encodes training samples to the latent space, along with a discriminator that is trained to discriminate on the basis of joint samples consisting of the data and the corresponding latent variable. In this architecture, the generator acted as a decoder in mapping a latent distribution to the data distribution. The authors demonstrated the algorithm's utility for semi-supervised classification by leveraging the inference machine instead of the discriminator. In their experiments, they were able to train ALI in an unsupervised manner on labeled as well as on unlabeled data, and then train an Support Vector Machine (SVM) as a classifier for the latent encodings on a subset of the labeled data. Thus, using such a technique, the authors were able to demonstrate the utility of an adversarial approach for SSL.

The success of these techniques has resulted in a wider implementation of the bidirectional architecture. Kumar et al. [40] proposed an "Augmented BiGAN" model inspired from the BiGAN and the ALI models. They argued that since the trained GANs produced realistic images, it could be assumed that the generator obtained the tangent space of an image's manifold. Therefore, they leveraged these tangents to inject desirable invariances into the classifier to improve its performance. This is in contrast to techniques that apply assumed invariances such as rotating and flipping. Furthermore, they proposed an improvement to the encoder presented by BiGAN that they claimed caused "class switching", which is when the generated data from an encoded latent space is of a different class to the original data. Therefore, they proposed a third input pair to feed to the discriminator consisting of a latent space derived from encoding a data point and the result of passing that encoded space through the generator. This pair would also be labeled as fake, and an additional loss term would be added to complement this change. Using such an approach, the authors reported a quantitative as well as qualitative improvement in performance as compared to BiGAN.

## 3.5. The TripleGAN Approach

Another class of techniques for the implementation of semi-supervised GANs is based on the TripleGAN architecture proposed by Li et al. [26]. Addressing the issue that the generator and discriminator cannot be optimal at the same time, this paper proposed a different approach to Dai et al. [29]. TripleGAN consisted of injecting an additional classifier, which along with the generator characterized the conditional distributions between images, while the discriminator was limited to identifying fake image–label pairs. Figure 4 shows the architecture used for the TripleGAN where the discriminator either outputs an accept (A) or a reject (R), which serve as the adversarial losses, while the classifier produces the cross-entropy losses (CE) for the supervised part of the learning.

The discriminator in TripleGAN takes image–label pairs of which there are 3 kinds; a true data–label pair from the labeled data (x,y), a generated data–label pair (G(z),y'), and an unlabeled data sample assigned a pair by passing it through the classifier ( $x_u$ ,P(c)) using pseudo-labeling.



Figure 4. The TripleGAN Architecture.

The resulting objective function is shown in Equation (4).

$$\min_{C,G} \max_{D} E_{(x,y)\sim p(x,y)}[\log D(x,y)] + \alpha E_{(x,y)\sim p_c(x,y)}[\log(1-D(x,y))] + (1-\alpha)E_{(x,y)\sim p_g(x,y)}[\log(1-D(G(y,z),y))]$$
(4)

EnhancedTGAN [34] was an extension of TripleGAN that redesigned the training targets of the generator and classifier. They designed the generator to produce images on the basis of a class distribution that was regulated by a feature-semantics matching term in the loss function. Furthermore, they added another classifier, which worked in collaboration to provide additional categorical information for the generator to train on.

Another notable extension of TripleGAN was MarginGAN [27] where the classifier increased the margin for real samples, and decreased the margin for fake samples. The generator tried to increase the margin for the fake samples only. This approach further helped prevent the drop in performance that typically happens due to the misclassification of a pseudo-label. They based their theory on [29] and aimed to implement the theory within the TripleGAN framework.

 $\Delta$ -GAN [35] combined ideas from BiGAN and TripleGAN. The model consisted of two generators and two discriminators, with the generators providing bidirectional mapping between domains and the discriminator classifying the real data pairs from the two kinds of fake data pairs.

Structured GANs [36] were similar to  $\Delta$ -GAN but assumed that generated data were conditioned on two independent latent variables, one of which encoded the designated semantics (*y*), while the other accounted for other factors of variation (*z*). Under the assumption that these latent variables were independent of each other, the authors proposed a set of two inference networks, one to map an input data point (*x*) to the designated semantics (*y*), and the other to map an input point to *z*. These networks were trained using two different adversarial games, one for each mapping of the input data.

 $R^3$ -CGAN [28] was another GAN architecture based on  $\Delta$ -GAN. This architecture was based on the observation that the classification network often gives incorrect yet confident predictions on unlabeled data while generating pseudo-labels. Furthermore, due to the imbalance between real and fake samples, the discriminator learns the real samples and rejects any unseen data even if they are real. The authors proposed using a regularization approach based on Random Regional Replacement in the learning process of the classification and discriminative networks. They implemented two discriminative networks in addition to the classifier and the generator. Fake sample pairs of two types were used, one consisting of synthesized data paired with the target label, and the other consisting of an unlabeled sample paired with its pseudo-label. One of the discriminators was trained to discriminate between real and fake images, while the other was trained to discriminate between two fake sample types.

EC-GAN [37] is another recent GAN using ideas from  $\Delta$ -GAN. In this architecture, a generator was trained to generate images, which were then instantly fed to the classifier that produced a pseudo-label. This combination of label and generated image was then used to train the classifier, with the loss function accounting for this semi-supervised loss

being multiplied by a hyperparameter that controlled how much importance the generated classification was given. The authors emphasized that the classifier was a separate network from the discriminator and empirically proved that it was a better approach as compared to a shared discriminator–classifier architecture. Furthermore, the use of CutMix [51] was noted as an augmentation strategy. The proposed architecture is shown in Figure 5.



Figure 5. The EC-GAN Architecture.

### 3.6. Manifold Regularization-Based Methods

Many recent GANs for SSL used manifold regularization. For example, Lecouat et al. [41] presented a methodology involving using the ability of GANs to model the manifold of natural images to perform manifold regularization by leveraging the Monte Carlo approximation of the Laplacian norm. They claimed that this regularization would encourage classifier invariance to local perturbations on the image as points close to the manifold would be assigned similar labels. For their work, the authors made use of the feature matching semi-supervised GAN presented in [32] as the base GAN. The primary challenge in this approach is the estimation of the Laplacian norm, for which they present an approach on the basis of the assumptions that GANs can model the distribution as well as the manifold of images. Based on these assumptions, their technique involved training the GAN on a large number of unlabeled images, after which they inferred that the GAN approximated the marginal distribution over images that could then be used to estimate the Laplacian norm over a classifier using Monte Carlo integrations with samples drawn from the space of latent representations of the generator. Furthermore, the second assumption allowed the manifold on the image space to be utilized to compute the gradient in the form of a Jacobian matrix with respect to the latent representations. Based on this, the classifier loss is shown in Equation (5)

$$\|f\|_{L}^{2} = \int_{x \in M} \|\nabla_{M} f(x)\|^{2} dP_{X}(x) \approx \frac{1}{n} \sum_{i=1}^{n} \|\nabla_{M} f(g(z^{(i)}))\|^{2} \approx \frac{1}{n} \sum_{i=1}^{n} \|J_{z} f(g(z^{(i)}))\|_{F}^{2}$$
(5)

A similar approach was taken by Lecouat et al. [41,42], where the Monte Carlo integrations were used to estimate a variant of the Laplacian norm seen in Equation (6).

$$\Omega(f) = \int_{x \in M} \|\nabla_M f\|_F dP_X$$
(6)

More recently, the SelfAttentionGAN [43] made use of manifold regularization as part of a self-attention mechanism for a semi-supervised GAN. A variable attention unit was used as part of the attention-based GAN architecture, while manifold regularization based on [42] was added as an additional regularization term to the loss function to make full use of unlabeled samples using a Monte Carlo approximation.

An interesting technique was proposed by SVMGAN [44], that tried to solve the issue of GAN-based SSL models being sensitive to local perturbations by introducing a

discriminator using a scalable support vector machine (SVM) classifier with manifold regularization, while SVM was used due to its nature of performing well in situations with small datasets, which fit the semi-supervised problem well. Furthermore, the use of manifold regularization was reported to force the discriminator to be resistant to local perturbations.

### 3.7. Two-GAN Approaches

MCGAN [45] attempted to solve the problem of GANs generalizing when two classes of images shared similar characteristics. In order to achieve this, a modification to the GAN training method was proposed. In this case, a number of classes have labels, while one class does not have labels. The approach suggested by the authors was to first separate the labeled classes from the unlabeled class, and then classify among the labeled classes. Two GANs were used with a training regime where the first discriminator was trained by passing images of first class labeled as real with the generator outputs labeled as fake. Furthermore, the authors passed images of the second class to the discriminator labeled as fake, which forced the generator to not generalize to the similar to the second class when learning features of the first class as the discriminator flagged any generated images bearing resemblance to second class as fake. The authors then used the variation score as proposed by AnoGAN [52] to classify a third class on the basis of the sum of the variation score so the two GANs (one trained on the first class and the other trained on second class). The architecture of the proposed GAN can be seen in Figure 6.



Figure 6. MCGAN Architecture.

Vanishing Twin GAN (VTGAN) [46] was an improvement over MCGAN, which heavily relied on labeled samples being used to train the discriminator and would fail in cases of semi-supervised learning where one of the classes did not have adequate labeled samples. The idea behind VTGAN was to train two GANs in parallel: a normal GAN to be used for classification, and a weak GAN to be used to improve the normal GAN's classification performance. The goal was to train the weak twin in such a way that the generator was stuck in the noisy image generation stage where it would not fall into modal collapse. The resulting noisy generation from this weak GAN was used as input to the normal twin with the fake labels. In order to weaken the weak twin, a number of strategies were used, such as making the network shallow, tuning the GAN's input noise dimension while decreasing the noise, and increasing strides of the transpose convolution and the max pooling layers.

A different approach using two GANs leveraged data augmentation in order to prepare a data augmentation GAN, which in turn was used to train another GAN. Inception-Augmentation GAN (IAGAN) [47] used augmentation of a given image in order to prepare the image to be used to train another GAN. The generator took in a batch of images and a Gaussian noise vector concatenated them after encoding the images using convolution and attention layers to a smaller dimension. A mix of inception and residual architectures was then used to enhance the generator's ability to capture details from the training space. The discriminator was simply a 4-layer CNN, which predicted whether an input image was a real image from the training data or an output of the generator. A generic objective function shown in Equation (7) was used.

$$\min_{G} \max_{D} V(D,G) = \mathbb{E}_{x \sim P_{data(x)}}[\log D(x)] + \mathbb{E}_{z \sim P_{z(z)}}[\log(1 - D(G(z)))]$$
(7)

# 3.8. GAN Using Stacked Discriminator

An interesting implementation involved leveraging the Conditional GANs [53] in a semi-supervised setting in a model called Semi-Supervised GAN (SS-GAN) [48]. The approach used gave the discriminator two tasks: detecting if a given image was real or fake and detecting whether a proposed attribute given to the image was real or fake. For the first task, both labeled and unlabeled samples were used in training; however, for the second task only the labeled images were used. In order to perform this task, a stacked discriminator approach was used with one discriminator for each task. Figure 7 shows the architecture of the SS-GAN and the flow of the training data, which makes use of both labeled and unlabeled images for the unsupervised discriminator and only labeled images for the supervised discriminator.



Figure 7. SS-GAN architecture.

### 4. Results

Tables 2–6 show lists of the sources reviewed and the techniques they set as the baseline comparison to their results. In order to provide ease of analysis, the different works reported are grouped as per the architecture followed by the work as discussed in the framework section.

Table 2. Baseline Models of Pseudo-labeling and Classifier approaches.

Citation	Authors	Date of Publication	Proposed Model	Baseline Models
[49]	Goodfellow et al.	June 2014	GAN (Original)	n/a
[30]	J. Springenberg	April 2016	CatGAN (Categorical)	MTC, PEA, PEA+, VAE + SVM, SS-VAE, Ladder T-model, Ladder-full
[32]	Salimans et al.	June 2016	Improved GAN	DGN, Virtual Adversarial, CatGAN, Skip Keep Generative Model, Ladder network, Auxiliary Deep Generative Model
[31]	A. Odena	October 2016	SGAN (Semi-Supervised)	CNN (isolated classifier, unspecified)
[29]	Dai et al.	November 2017	GoodBadGAN	CatGAN, SDGM, Ladder network, ADGM, FM, ALI, VAT small, TripleGAN, П model, VAT + EntMin + Large

[43]

A. Haque

March 2021

Citation	Authors	Date of Publication	<b>Proposed Model</b>	<b>Baseline Models</b>	
[19]	Wei et al.	March 2018	CT-GAN	Ladder, VAT, CatGAN, Improved GAN, TripleGAN	
[33]	Sun et al.	October 2020	MatchGAN	StarGAN	
		Table 3. Baseline Mo	dels of Encoder-based Approa	ches.	
Citation	Authors	Date of Publication	Proposed Model	<b>Baseline Models</b>	
[38].	Donahue et al.	May 2016	BiGAN	-	
[39]	Dumoulin et al.	February 2017	ALI (Adversarially Learned Inference)	CIFAR-10: Ladder network, CatGAN, GAN (Salimans 2016); SVHN: VAE, SWWAE, DCGAN + L2SVM, SDGM, GAN (Salimans 2016)	
[40]	Kumar et al.	December 2017	Augmented BiGAN		
	Table 4. Baseline Models for TripleGAN implementations.				
Citation	Authors	Date of Publication	Proposed Model	<b>Baseline Models</b>	
[26]	Li et al.	November 2017	TripleGAN	M1 + M2, VAT, Ladder, Conv-Ladder, ADGM, SDGM, MMCVA, CatGAN, Improved GAN, ALI	
[35]	Gan et al.	November 2017	TriangleGAN	CatGAN, Improved GAN, ALI, TripleGAN	
[36]	Deng et al.	November 2017	SGAN (Structured)	Ladder, VAE, CatGAN, ALI, Improved GAN, TripleGAN	
[27]	J. Dong and T. Lin	November 2019	MarginGAN	NN, SVM, CNN, TSVM, DBN-rNCA, EmbedNN, CAE, MTC	
[34]	Wu et al.	January 2020	EnhancedTGAN (Triple)	Ladder network, SPCTN, П model, Temporal Ensembling, Mean Teacher, VAT, VAdD, VAdD + VAT, SNTG + П model, SNTG + VAT, CatGAN, Improved GAN, ALI, TripleGAN, GoodBadGAN, CT-GAN, TripleGAN	
[28]	Liu et al.	August 2020	R <sup>3</sup> -CGAN (Random Regional Replacement Class-Conditional)	Ladder network, SPCTN, II model, Temporal Ensembling, Mean Teacher, VAT, VAdD, SNTG + II model, Deep Co-Train, CCN, ICT, CatGAN, Improved GAN, ALI, TripleGAN, Triangle-GAN, GoodBadGAN, CT-GAN, EnhancedTGAN	

# Table 2. Cont.

 Table 5. Baseline Models for Manifold Regularization-based approaches.

EC-GAN (External Classifier)

DCGAN

Citation	Authors	Date of Publication	Proposed Model	<b>Baseline Models</b>
[41]	Lecouat et al.	May 2018	Laplacian-based GAN	Ladder network, II model, VAT, VAT + EntMin, CatGAN, Improved GAN, TripleGAN, Improved semi-GAN, Bad GAN
[42]	Lecouat et al.	July 2018	Monte Carlo-based GAN	П model, Mean Teacher, VAT, Vat + EntMin, Improved GAN, Improved Semi-GAN, ALI, TripleGAN, Bad GAN, Local GAN
[43]	Xiang et al.	November 2019	SelfAttentionGAN	CatGAN, Improved GAN, TripleGAN, Bad GAN, Local GAN, Manifold-GAN, CT-GAN, Ladder network, π-model, Temporal Ensembling w/augmentation, VAT + EntMin w/ aug, MeanTeacher, MeanTeacher w/aug, VAT + Ent + SNGT w/aug
[44]	Tang et al.	August 2020	SSVM-GAN (Scalable SVM)	Ladder Network, CatGAN, ALI, VAT, FM GAN, Improved FM, GAN, TripleGAN, II model, Bad GAN

Citation	Authors	Date of Publication	Proposed Model	Baseline Models
[48]	Sricharan et al.	August 2017	SS-GAN (Semi-Supervised)	C-GAN (conditional GAN on full dataset), SC-GAN (conditional GAN only on labeled dataset), AC-GAN (supervised auxiliary classifier GAN on full dataset), SA-GAN (semi-supervised AC-GAN)
[47]	Motamed et al.	January 2021	IAGAN (Inception-Augmentation)	AnoGAN, AnoGAN w/traditional augmentation, DCGAN
[45]	S. Motamed and F. Khalvati	February 2021	MCGAN (Multi-Class)	DCGAN
[46]	S. Motamed and F. Khalvati	March 2021	VTGAN (Vanishing Twin)	OC-SVM, IF, AnoGAN, NoiseGAN, Deep SVDD

Table 6. Baseline Models for Assorted Approaches.

The discussed works were analyzed in terms of the results reported by the authors for their respective proposed models and chronologically summarized in Tables 7–11 where the proposed model, evaluation datasets, and the results are detailed.

Table 7. Pseudo-labeling and Classifier Approaches Results Summary.

Citation	Proposed Model	Datasets Evaluated On	Results
[49]	GAN (Original)	MNIST, TFD	Gaussian Parzen window: MNIST: 225, TDF: 2057
[30]	CatGAN (Categorical)	MNIST	1.91% PI-MNIST test error w/100 labeled examples, outperforms all models except Ladder-full (1.13%)
[32]	Improved GAN	MNIST, CIFAR-10, SVHN	MNIST: 93 incorrectly predicted test examples w/ 100 labeled samples, outperforms all other; CIFAR-10: 18.63 test error rate w/4000 labeled samples, outperforms all other; SVHN: 8.11% incorrectly predicted test examples w/1000 labeled samples, outperforms all other
[31]	SGAN (Semi-Supervised)	MNIST	96.4% classifier accuracy w/1000 labeled samples, comparable to isolated CNN classifier (96.5%)
[29]	GoodBadGAN	MNIST, SVHN, CIHAR-10	MNIST: 79.5 # of errors, outperforms all; SVHN: 4.25% errors, outperforms all; CIFAR-10: 14.41% errors, outperforms all except Vat + EntMin + Large
[19]	CT-GAN	MNIST	0.89% error rate, outperformed all
[33]	MatchGAN	CelebA, RaFD	(For both datasets, 20% of training data labeled) CelebA: 6.34 FID, 3.03 IS; RaFD: 9.94 FID, 1.61 IS; outperformed StarGAN in all metrics

# Table 8. Encoder-based Approaches Results Summary.

Citation	Proposed Model	Datasets Evaluated On	Results
[38]	BiGAN	ImageNet	Max Classification accuracy: 56.2% with conv classifier
[39]	ALI (Adversarially Learned Inference)	CIFAR-10, SVHN, CelebA, ImageNet (center-cropped 64 × 64 version)	CIFAR-10: 17.99 misclassification rate w/4000 labeled samples, outperforms all; SVHN: 7.42 misclassification rate w/1000 labeled samples, outperforms all
[40]	Augmented BiGAN	SVHN, CIFAR-10	SVHN: 4.87 test error w/500 labeled, 4.39 test error w/1000 labeled, outperforms all for both; CIFAR-10: 19.52 test error w/1000 labeled, outperforms all, 16.20 test error w/4000 labeled, outperforms all except Temporal Ensembling

Citation	Proposed Model	Datasets Evaluated On	Results
[26]	TripleGAN	MNIST, SVHN, and CIFAR-10	MNIST: 0.91% error rate w/100 labeled samples, outperforms all except Conv-Ladder; SVHN: 5.77% error rate w/1000 labeled samples, outperforms all except MMCVA; CIFAR-10: 16.99% error rate w/4000 labeled samples, outperforms all
[35]	Triangle-GAN	CIFAR-10	16.80% error rate w/4000 labeled samples, outperforms all

Citation	Proposed Model	Datasets Evaluated On	Results
[36]	SGAN (Structured)	MNIST, SVHN, CIFAR-10	MNIST: 0.89% error rate w/100 labeled, outperforms all but equal as Ladder; SVHN: 5.73% error rate w/1000 labeled, outperforms all; CIFAR-10: 17.26% error rate w/4000 labeled, outperforms all
[27]	MarginGAN	MNIST	2.06% error rate w/3000 labels, outperformed all
[34]	EnhancedTGAN (Triple)	MNIST, SVHN, CIFAR-10	MNIST: 0.42% error rate w/100 labels, outperforms all; SVHN: 2.97% error rate w/1000 labels, outperforms all; CIFAR-10: 9.42% error rate w/4000 labels, outperforms all
[28]	R <sup>3</sup> -CGAN (Random Regional Replacement Class-Conditional)	SVHN, CIFAR-10	SVHN: 2.79% error rate w/1000 labels, outperformed all except equal with EnhancedTGAN; CIFAR-10: 6.69% error rate w/4000 labels, outperformed all
[43]	EC-GAN (External Classifier)	SVHN, X-ray Dataset	SVHN: 93.93% accuracy w/25% of dataset, outperformed DCGAN; X-ray: 96.48% accuracy w/25% of dataset, outperformed DCGAN

# Table 9. Cont.

### Table 10. Manifold Regularization-based approaches Results Summary.

Citation	Proposed Model	Datasets Evaluated On	Results
[41]	Laplacian-based GAN	SVHN, CIFAR-10	SVHN: 4.51% error rate w/1000 labeled, outperformed all except Vat + EntMin, Improved semi-GAN, and Bad GAN; CIFAR-10: 14.45% error rate w/4000 labeled, outperformed all except Vat + EntMin and Bad GAN
[42]	Monte Carlo-based GAN	CIFAR-10, SVHN	CIFAR-10: 14.34% error rate w/4000 labels, outperformed all except VAT, VAT + EntMin, and Local GAN; SVHN: 4.63% error rate w/1000 labels, outperformed VAT + EntMin and Improved semi-GAN
[43]	SelfAttentionGAN	SVHN, CIFAR-10	CIFAR-10: 9.87% error rate w/4000 labels, outperformed all; SVHN: 4.30% error rate w/1000 labels, outperformed all except Bad GAN, VAT + EntMin w/aug, MeanTeacher w/aug, VAT + Ent + SNGT w/aug
[44]	SSVM-GAN (Scalable SVM)	CIFAR-10, SVHN	CIFAR-10: 14.27% error rate w/4000 labels, outperformed all; SVHN: 4.54% error rate w/1000 labels, outperformed all except Bad GAN

# Table 11. Assorted Approaches Results Summary.

Citation	Proposed Model	Datasets Evaluated on	Results
[48]	SS-GAN (Semi-Supervised)	MNIST, CelebA, CIFAR-10	MNIST: 0.1044 class prediction error, outperforms only SA-GAN, 0.0160 reconstruction error, outperforms SA-GAN and SC-GAN (both metrics w/20 labeled samples); CelebA: 0.040 reconstruction error, outperforms all except C-GAN; CIFAR-10: 0.299 class pred error, outperforms only AC-GAN and SC-GAN, 0.061 recon error, outperforms all except C-GAN
[47]	IAGAN (Inception-Augmentation)	Pneumonia X-rays: Dataset I (3765 imgs), Dataset II (4700 imgs)	Dataset I: 0.90 AUC, outperformed all; Dataset II: 0.76 AUC, outperformed all
[45]	MCGAN (Multi-Class)	MNIST, F-MNIST	MNIST: 0.9 AUC unknown class classification and 0.84 known class classification, outperformed DCGAN; F-MNIST: 0.79 AUC unknown & 0.65 known, outperformed DCGAN
[46]	VTGAN (Vanishing Twin)	MNIST, F-MNIST	MNIST: 0.90, 0.92, 0.85, and 0.86 AUC, outperformed all in all 4 experiments; F-MNIST: 0.87, 0.76, 0.70, 0.57, 0.62, 0.70 AC, outperformed all in 4 out of 6 experiments

# 5. Discussion

# 5.1. Quantitative Analysis

A number of GANs were chosen as representative models from each technique. Cat-GAN was chosen from the initial implementations. Similarly, ALI was the model of choice for comparison among the encoder-based architectures, and TripleGAN was used as the baseline of choice for its class of models. Table 12 displays a summary of notable works across categories and their results in order to enable a deeper comparison.

Citation	Category	Proposed Model	Results
[30]	Pseudo-labeling and Classifiers	CatGAN	1.91% PI-MNIST test error w/100 labeled examples, outperforms all models except Ladder-full (1.13%)
[39]	Encoder-based	ALI	CIFAR-10: 17.99 misclassification rate w/4000 labeled samples, outperforms all; SVHN: 7.42 misclassification rate w/1000 labeled samples, outperforms all
[26]	TripleGAN	TripleGAN	MNIST: 0.91% error rate w/100 labeled samples, outperforms all except Conv-Ladder; SVHN: 5.77% error rate w/1000 labeled samples, outperforms all except MMCVA; CIFAR-10: 16.99% error rate w/4000 labeled samples, outperforms all
[44]	Manifold Regularization	SSVM-GAN	CIFAR-10: 14.27% error rate w/4000 labels, outperformed all; SVHN: 4.54% error rate w/1000 labels, outperformed all except Bad GAN
[28]	TripleGAN	R <sup>3</sup> -CGAN	SVHN: 2.79% error rate w/1000 labels, outperformed all except equal with EnhancedTGAN; CIFAR-10: 6.69% error rate w/4000 labels, outperformed all

Table 12. Summary of notable works.

A natural progression can be seen where encoder-based architectures such as ALI outperformed CatGAN, and this, in turn, was outperformed by TripleGAN and its derivatives. The more recent manifold regularization-based approaches also outperformed TripleGAN. However, the most recent manifold regularization-based paper [44] reported a 4.54% error rate on SVHN using 1000 labeled samples and a 14.27% error rate on CIFAR-10 using 4000 labeled samples. This model was outperformed by the most recent TripleGAN-based approach [28], which reported a 2.79% error rate on SVHN and 6.69% error rate on CIFAR for the same amount of labeled samples. Therefore, it is reasonable to claim that the R<sup>3</sup>-CGAN architecture holds the current state of the art as none of the other papers surveyed had a similar evaluation process or a comparison to this model. A number of interesting aspects of the R<sup>3</sup>-CGAN that could have contributed to its success. While the underlying architecture was based on TripleGAN, a Random Regional Replacement regularization was applied by making use of the CutMix mix-sample augmentation technique [51]. This technique has been implemented in non-generative semi-supervised learning techniques in order to achieve consistency regularization with good results. Therefore, its success in a generative architecture suggests adaptation of other semi-supervised learning techniques into GANs as well. It is interesting to note that while R<sup>3</sup>-CGAN is seemingly the best performing GAN-based technique currently available, it fades in comparison to non-GAN state of the art SSL techniques such as FixMatch [8], which reported error rates of 4.26% on CIFAR-10 with 4000 labeled samples and 2.28% on SVHN with 1000 labeled samples, in addition to showing a good performance of 11.39% error for CIFAR-10 with only 40 labeled samples and 3.96% for SVHN with 40 labeled samples. Therefore, the gap between GANbased SSL and other state of the art techniques is apparent, and so it would be interesting to attempt to apply some of the techniques used in other SSL algorithms to GANs in order to unify the enhanced performance seen in the state of the art SSL algorithms with the generative aspect that GANs are known for.

### 5.2. Qualitative Analysis

The initial approaches involving the implementation of pseudo-labeling, and an addition of a classifier had the advantage of being simple to implement without additional heavy computational load. These techniques, however, were limited in performance, where other more complex techniques were seen to outperform this class of methods. Encoderbased techniques were introduced with the intention of leveraging the feature space in the training of the models. However, success of these techniques was dependent on the latent representations being representative of the classification task at hand and, therefore, an assumption that could vary based on the target domain. Additionally, the addition of an encoder resulted in increased computational requirements for the training, which might be limited based on the available hardware resources. Conditional approaches involved discrimination in pairs of data points and labels with the classifier acting as a third player, which has been seen as a good solution to help resolve the conflict between having a good generator and a good classifier. However, the reliance on the class label as an input to the discriminator can be a point of failure in cases where the class distribution in the dataset is imbalanced to the extent that the discriminator only learns the majority classes to be real. In such a case, the generator will be strongly biased towards the majority class. A number of recent approaches have used manifold regularization to ensure that the model remains resistant to perturbations to input samples. However, such approaches rely on the assumption that any unseen data will lie on the same manifold as the perturbations used to perform the manifold regularization, which might fail in some cases based on the application domain.

An interesting difference in approaches is also observed among the various techniques analyzed in terms of the training objectives. While one class of techniques focused on a strategy where the generator was weakened to boost the discriminator (e.g., GoodBadGAN [29]), a different class of techniques leveraged a good generator to boost the performance of the classifier (e.g., TripleGAN [26]). Li et al. [54] conducted a comparative analysis of these two techniques by training the GoodBadGAN as the BadGAN approach and the TripleGAN as the GoodGAN approach on the MNIST, CIFAR10, and SVHN benchmark datasets with a varying level of labeled samples. Their conclusion was that while GoodBadGAN outperformed TripleGAN when there were a medium number of labeled samples, TripleGAN performed better with less data, thus demonstrating a lack of sensitivity to the number of labeled samples. Furthermore, the authors also provided visualizations for the images generated in the case of both of the techniques. Figure 8 is reproduced with permission from [54] and displays the generated images for both models.



Figure 8. Examples of images generated by good and bad GANs [55].

As can be seen in Figure 8, in the case of the GoodBadGAN, the images produced by the generator were far from ideal, indeed confusing the digits in the case of MNIST while

failing entirely in the case of SVHN and CIFAR. The TripleGAN (GoodGAN) architecture, however, was able to produce clear distinct images while also performing well for lower amounts of unlabeled data. The authors suggested that future work could involve both types of architectures being used complimentarily.

## 6. Future Directions

A number of future research directions can be explored. One direction is in terms of the model architecture, and training methodology itself. With the success of R<sup>3</sup>-CGAN's usage of CutMix, an interesting direction for research could be the implementation of further semi-supervised methods alongside GANs. While this is not a new concept and work including Chen et al. [20] have previously used SSL algorithms such as MeanTeacher to achieve consistency regularization, however, newer SSL techniques could also be looked into, as well. For example, the idea of automated augmentation techniques like RandAugment [55] and AutoAugment [56] used by state of the art SSL techniques like UDA [57] and FixMatch [8] can be explored. Another interesting direction could be the unifying of the current dominant GAN-based SSL techniques by adding manifold regularization to the  $R^3$ -CGAN implementation of TripleGAN. Since both techniques have the best results in recent works, combining them could be a step forward in the area of GAN-based SSL. On a similar note, future work can be carried out towards unifying the contrasting approaches of preparing a bad GAN for classification with approaches aiming to simultaneously improve both aspects of the GAN. This is a promising direction as BadGAN approaches have been noted to perform better for larger amounts of data, while GoodGAN approaches have outperformed for smaller levels of data. A unified method would be able to take advantage of these to form a more robust model.

Finally, attempts at training using a lower number of labeled samples could be undertaken in an effort to mimic state of the art SSL techniques, and to obtain a baseline for the current GAN-based performance for situations where an extremely low number of labeled samples are present. Consequently, efforts can be made to investigate the performance of existing techniques when implemented on real-world applications across domains, many of which have their own unique peculiarities. An example of such characteristics is seen in situations where the data relevant to the domain consists of a class imbalance, with the class of interest often being in the minority, such as in applications that include disease or fraud detection [58]. Investigations into how the existing solutions perform in these real-life domains will establish their viability and, in turn, can serve to further the field and improve the collective performance of the semi-supervised learning techniques.

# 7. Conclusions

Given the increasing interest in the field of semi-supervised learning, and the rapid progress being made in generative learning, a survey was conducted to analyze recent research in using GANs for semi-supervised learning. The previous work was catagoized based on the advancement being proposed, the model architecture, and the training procedures. Furthermore, the approach followed by each paper was discussed before a quantitative analysis was conducted based on the performance obtained by each of the works in their experimentation. Finally, a qualitative analysis of the various categories was also conducted to better understand the advantages and disadvantages of the diverse approaches, after which a number of possible directions for future work were identified in order to encourage advances in the field of using generative adversarial networks for semi-supervised learning.

**Author Contributions:** Conceptualization, A.R.S. and I.Z.; methodology, A.R.S.; formal analysis, A.R.S.; investigation, A.R.S.; resources, A.R.S.; writing—original draft preparation, A.R.S.; writing—review and editing, I.Z.; visualization, A.R.S.; supervision, I.Z. All authors have read and agreed to the published version of the manuscript.

**Funding:** The work in this paper was supported, in part, by the Open Access Program from the American University of Sharjah [grant number: OAPCEN-1410-E00027].

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Acknowledgments: This paper represents the opinions of the authors and does not mean to represent the position or opinions of the American University of Sharjah.

Conflicts of Interest: The authors declare no conflict of interest.

# References

- 1. van Engelen, J.E.; Hoos, H.H. A Survey on Semi-Supervised Learning. Mach. Learn. 2020, 109, 373–440. [CrossRef]
- Károly, A.I.; Fullér, R.; Galambos, P. Unsupervised Clustering for Deep Learning: A Tutorial Survey. Acta Polytech. Hung. 2018, 15, 29–53.
- 3. Ouali, Y.; Hudelot, C.; Tami, M. An Overview of Deep Semi-Supervised Learning. arXiv 2020, arXiv:2006.05278.
- 4. Yang, X.; Song, Z.; King, I.; Xu, Z. A Survey on Deep Semi-Supervised Learning. *arXiv* **2021**, arXiv:2103.00550.
- Chapelle, O.; Zien, A. Semi-Supervised Classification by Low Density Separation. In Proceedings of the Tenth International Workshop on Artificial Intelligence and Statistics, AISTATS 2005, Bridgetown, Barbados, 6–8 January 2005; Cowell, R.G., Ghahramani, Z., Eds.; PMLR: London, UK; Volume R5, pp. 57–64.
- 6. Krizhevsky, A. Learning Multiple Layers of Features from Tiny Images; Technical report; University of Toronto: Toronto, ON, Canada, 2009.
- Netzer, Y.; Wang, T.; Coates, A.; Bissacco, A.; Wu, B.; Ng, A.Y. Reading Digits in Natural Images with Unsupervised Feature Learning. In Proceedings of the NIPS Workshop on Deep Learning and Unsupervised Feature Learning, Granada, Spain, 12–17 December 2011.
- 8. Sohn, K.; Berthelot, D.; Li, C.-L.; Zhang, Z.; Carlini, N.; Cubuk, E.D.; Kurakin, A.; Zhang, H.; Raffel, C. FixMatch: Simplifying Semi-Supervised Learning with Consistency and Confidence. *arXiv* 2020, arXiv:2001.07685.
- Oliver, A.; Odena, A.; Raffel, C.; Cubuk, E.D.; Goodfellow, I.J. Realistic Evaluation of Deep Semi-Supervised Learning Algorithms. In Proceedings of the 32nd International Conference on Neural Information Processing Systems, Montréal, QC, Canada, 3–8 December 2018; Curran Associates Inc.: Red Hook, NY, USA, 2018; pp. 3239–3250.
- Zhang, H.; Zhang, Z.; Odena, A.; Lee, H. Consistency Regularization for Generative Adversarial Networks. In Proceedings of the International Conference on Learning Representations, Shenzhen, China, 15–17 February 2020.
- Sajjadi, M.; Javanmardi, M.; Tasdizen, T. Regularization with Stochastic Transformations and Perturbations for Deep Semi-Supervised Learning. In Proceedings of the 30th International Conference on Neural Information Processing Systems, Barcelona, Spain, 5–10 December 2016; Curran Associates Inc.: Red Hook, NY, USA, 2016; pp. 1171–1179.
- 12. Laine, S.; Aila, T. Temporal Ensembling for Semi-Supervised Learning. arXiv 2017, arXiv:1610.02242.
- Tarvainen, A.; Valpola, H. Mean Teachers Are Better Role Models: Weight-Averaged Consistency Targets Improve Semi-Supervised Deep Learning Results. In Proceedings of the 31st International Conference on Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017; Curran Associates Inc.: Red Hook, NY, USA, 2017; pp. 1195–1204.
- Luo, Y.; Zhu, J.; Li, M.; Ren, Y.; Zhang, B. Smooth Neighbors on Teacher Graphs for Semi-Supervised Learning. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 8896–8905.
- Miyato, T.; Maeda, S.; Koyama, M.; Ishii, S. Virtual Adversarial Training: A Regularization Method for Supervised and Semi-Supervised Learning. *IEEE Trans. Pattern Anal. Mach. Intell.* 2019, 41, 1979–1993. [CrossRef] [PubMed]
- 16. Park, S.; Park, J.-K.; Shin, S.-J.; Moon, I.-C. Adversarial Dropout for Supervised and Semi-Supervised Learning. *arXiv* 2017, arXiv:1707.03631.
- Verma, V.; Lamb, A.; Kannala, J.; Bengio, Y.; Lopez-Paz, D. Interpolation Consistency Training for Semi-Supervised Learning. In Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, Macao, China, 10–16 August 2019; International Joint Conferences on Artificial Intelligence Organization: Macao, China, 2019; pp. 3635–3641.
- Berthelot, D.; Carlini, N.; Cubuk, E.D.; Kurakin, A.; Sohn, K.; Zhang, H.; Raffel, C. ReMixMatch: Semi-Supervised Learning with Distribution Alignment and Augmentation Anchoring. *arXiv* 2020, arXiv:1911.09785.
- 19. Wei, X.; Gong, B.; Liu, Z.; Lu, W.; Wang, L. Improving the Improved Training of Wasserstein GANs: A Consistency Term and Its Dual Effect. *arXiv* **2018**, arXiv:1803.01541.
- 20. Chen, Z.; Ramachandra, B.; Vatsavai, R.R. Consistency Regularization with Generative Adversarial Networks for Semi-Supervised Learning. *arXiv* 2020, arXiv:2007.03844.
- 21. Zhao, Z.; Singh, S.; Lee, H.; Zhang, Z.; Odena, A.; Zhang, H. Improved Consistency Regularization for GANs. *arXiv* 2020, arXiv:2002.04724.
- 22. Lee, D.-H. Pseudo-Label: The Simple and Efficient Semi-Supervised Learning Method for Deep Neural Networks. In Proceedings of the Workshop on Challenges in Representation Learning, Atlanta, GA, USA, 21 June 2013; Volume 3.

- Shi, W.; Gong, Y.; Ding, C.; Ma, Z.; Tao, X.; Zheng, N. Transductive Semi-Supervised Deep Learning Using Min-Max Features. In Proceedings of the Computer Vision—ECCV 2018, Munich, Germany, 8–14 September 2018; Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y., Eds.; Springer International Publishing: Cham, Switzerland, 2018; pp. 311–327.
- Iscen, A.; Tolias, G.; Avrithis, Y.; Chum, O. Label Propagation for Deep Semi-Supervised Learning. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019; pp. 5065–5074.
- 25. Arazo, E.; Ortego, D.; Albert, P.; O'Connor, N.E.; McGuinness, K. Pseudo-Labeling and Confirmation Bias in Deep Semi-Supervised Learning. *arXiv* 2020, arXiv:1908.02983.
- 26. Li, C.; Xu, K.; Zhu, J.; Zhang, B. Triple Generative Adversarial Nets. arXiv 2017, arXiv:1703.02291.
- Dong, J.; Lin, T. MarginGAN: Adversarial Training in Semi-Supervised Learning. In Proceedings of the Advances in Neural Information Processing Systems, Vancouver, BC, Canada, 8–14 December 2019; Wallach, H., Larochelle, H., Beygelzimer, A., Alché-Buc, F., Fox, E., Garnett, R., Eds.; Curran Associates, Inc.: Red Hook, NY, USA, 2019; Volume 32.
- Liu, Y.; Deng, G.; Zeng, X.; Wu, S.; Yu, Z.; Wong, H.-S. Regularizing Discriminative Capability of CGANs for Semi-Supervised Generative Learning. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020; pp. 5719–5728.
- Dai, Z.; Yang, Z.; Yang, F.; Cohen, W.W.; Salakhutdinov, R. Good Semi-Supervised Learning That Requires a Bad GAN. *arXiv* 2017, arXiv:1705.09783.
- Springenberg, J.T. Unsupervised and Semi-Supervised Learning with Categorical Generative Adversarial Networks. arXiv 2016, arXiv:1511.06390.
- 31. Odena, A. Semi-Supervised Learning with Generative Adversarial Networks. arXiv 2016, arXiv:1606.01583.
- Salimans, T.; Goodfellow, I.; Zaremba, W.; Cheung, V.; Radford, A.; Chen, X. Improved Techniques for Training GANs. arXiv 2016, arXiv:1606.03498.
- 33. Sun, J.; Bhattarai, B.; Kim, T.-K. MatchGAN: A Self-Supervised Semi-Supervised Conditional Generative Adversarial Network. *arXiv* 2020, arXiv:2006.06614.
- Wu, S.; Deng, G.; Li, J.; Li, R.; Yu, Z.; Wong, H.-S. Enhancing TripleGAN for Semi-Supervised Conditional Instance Synthesis and Classification. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019; pp. 10083–10092.
- 35. Gan, Z.; Chen, L.; Wang, W.; Pu, Y.; Zhang, Y.; Liu, H.; Li, C.; Carin, L. Triangle Generative Adversarial Networks. *arXiv* 2017, arXiv:1709.06548.
- 36. Deng, Z.; Zhang, H.; Liang, X.; Yang, L.; Xu, S.; Zhu, J.; Xing, E.P. Structured Generative Adversarial Networks. *arXiv* 2017, arXiv:1711.00889.
- 37. Haque, A. EC-GAN: Low-Sample Classification Using Semi-Supervised Algorithms and GANs. arXiv 2021, arXiv:2012.15864.
- 38. Donahue, J.; Krähenbühl, P.; Darrell, T. Adversarial Feature Learning. *arXiv* **2017**, arXiv:1605.09782.
- 39. Dumoulin, V.; Belghazi, I.; Poole, B.; Mastropietro, O.; Lamb, A.; Arjovsky, M.; Courville, A. Adversarially Learned Inference. *arXiv* **2017**, arXiv:1606.00704.
- Kumar, A.; Sattigeri, P.; Fletcher, P.T. Semi-Supervised Learning with GANs: Manifold Invariance with Improved Inference. *arXiv* 2017, arXiv:1705.08850.
- 41. Lecouat, B.; Foo, C.-S.; Zenati, H.; Chandrasekhar, V.R. Semi-Supervised Learning with GANs: Revisiting Manifold Regularization. *arXiv* 2018, arXiv:1805.08957.
- 42. Lecouat, B.; Foo, C.-S.; Zenati, H.; Chandrasekhar, V. Manifold Regularization with GANs for Semi-Supervised Learning. *arXiv* 2018, arXiv:1807.04307.
- Xiang, X.; Yu, Z.; Lv, N.; Kong, X.; Saddik, A.E. Attention-Based Generative Adversarial Network for Semi-Supervised Image Classification. *Neural Process. Lett.* 2020, 51, 1527–1540. [CrossRef]
- Tang, X.; Yu, X.; Xu, J.; Chen, Y.; Wang, R. Semi-Supervised Generative Adversarial Networks Based on Scalable Support Vector Machines and Manifold Regularization. In Proceedings of the 2020 Chinese Control And Decision Conference (CCDC), Hefei, China, 22–24 August 2020; pp. 4264–4269.
- 45. Motamed, S.; Khalvati, F. Multi-Class Generative Adversarial Nets for Semi-Supervised Image Classification. *arXiv* 2021, arXiv:2102.06944.
- Motamed, S.; Khalvati, F. Vanishing Twin GAN: How Training a Weak Generative Adversarial Network Can Improve Semi-Supervised Image Classification. *arXiv* 2021, arXiv:2103.02496.
- 47. Motamed, S.; Rogalla, P.; Khalvati, F. Data Augmentation Using Generative Adversarial Networks (GANs) for GAN-Based Detection of Pneumonia and COVID-19 in Chest X-Ray Images. *arXiv* **2021**, arXiv:2006.03622. [CrossRef] [PubMed]
- 48. Sricharan, K.; Bala, R.; Shreve, M.; Ding, H.; Saketh, K.; Sun, J. Semi-Supervised Conditional GANs. arXiv 2017, arXiv:1708.05789.
- Goodfellow, I.J.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y. Generative Adversarial Networks. *arXiv* 2014, arXiv:1406.2661. [CrossRef]
- 50. Arjovsky, M.; Chintala, S.; Bottou, L. Wasserstein GAN. arXiv 2017, arXiv:1701.07875.
- Yun, S.; Han, D.; Oh, S.J.; Chun, S.; Choe, J.; Yoo, Y. CutMix: Regularization Strategy to Train Strong Classifiers with Localizable Features. arXiv 2019, arXiv:1905.04899.

- 52. Schlegl, T.; Seeböck, P.; Waldstein, S.M.; Schmidt-Erfurth, U.; Langs, G. Unsupervised Anomaly Detection with Generative Adversarial Networks to Guide Marker Discovery. *arXiv* **2017**, arXiv:1703.05921.
- 53. Mirza, M.; Osindero, S. Conditional Generative Adversarial Nets. arXiv 2014, arXiv:1411.1784.
- 54. Li, W.; Wang, Z.; Li, J.; Polson, J.; Speier, W.; Arnold, C. Semi-Supervised Learning Based on Generative Adversarial Network: A Comparison between Good GAN and Bad GAN Approach. *arXiv* **2019**, arXiv:1905.06484.
- 55. Cubuk, E.D.; Zoph, B.; Shlens, J.; Le, Q.V. RandAugment: Practical Automated Data Augmentation with a Reduced Search Space. *arXiv* **2019**, arXiv:1909.13719.
- Cubuk, E.D.; Zoph, B.; Mané, D.; Vasudevan, V.; Le, Q.V. AutoAugment: Learning Augmentation Strategies From Data. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019; pp. 113–123.
- 57. Xie, Q.; Dai, Z.; Hovy, E.; Luong, M.-T.; Le, Q.V. Unsupervised Data Augmentation for Consistency Training. *arXiv* 2020, arXiv:1904.12848.
- 58. Johnson, J.M.; Khoshgoftaar, T.M. Survey on Deep Learning with Class Imbalance. J. Big Data 2019, 6, 27. [CrossRef]