Inducing Self-Supervised Learning in GANs

Sanidhya Mangal
Vanderbilt University
sanidhya.v.mangal@vanderbilt.edu

Daniel Shu
Vanderbilt University
daniel.m.shu@vanderbilt.edu

Rishav Sen
Vanderbilt University
rishav.sen@vanderbilt.edu

Jatin Krishna Sai Kodali
Vanderbilt University
jatin.krishna.sai.kodali@vanderbilt.edu

Abstract

Generative Adversarial Networks (GANs) are one of the most popular classes of generative models. Vanilla GANs suffer from many drawbacks, which has led to exploration in several variations. One key issue is the lack of stability in GANs, caused by the non-stationary space the discriminator learns in as a of the adversarial game. We mitigate this problem by inducing self-supervised learning in a GAN and adding a self-supervised loss component to the discriminator. The self-supervised head aids the discriminator in learning useful feature representations that are not forgotten in subsequent training epochs. Our work focuses on exploring two potential self-supervised tasks, rotational and contrastive learning on images. We mainly test the quality of the generated image samples using FID based scores. We show that our method was able to improve upon Vanilla GANs, with the Contrastive Learning GAN performing the best.

1 Introduction

Generative modelling is a thoroughly established paradigm of machine learning that uses input training samples to learn a statistical model that represents the underlying distribution and can generate new samples. Popular examples are Gaussian mixture models and variational autoencoders. We focus on generative adversarial networks (GANs), a powerful class of generative models that are used in numerous applications. GANs augment more traditional generative models by introducing a discriminative model that evaluates the generator; both networks then learn their respective tasks through adversarial training. However, this results in instability of the discriminator because it is trained partially on the output distribution of the generator which updates and changes with each iteration. The effectiveness of the adversarial training relies on both networks to perform well, so the generator is also heavily affected.

We thought to address this issue by utilizing self-supervised learning. By applying this technique to the discriminator, it gains a new loss component that also induces the network to learn feature representations that improve its accuracy in the classification task. The main benefit of adding this component is that it only takes the real distribution as its input. Thus, the discriminator is less susceptible to shifts in the generated distribution and learns more stable representations of the data. The advantage of self-supervised learning compared to other types of representational learning that we could have considered is that it does not require any additional data or labels that cannot be generated easily by augmenting the existing data set.

We present a GAN on image inputs with a discriminator designed based on those ideas. It is composed of a feature learner with a CNN backbone followed by two heads of multi-layer perceptrons (MLPs): one to classify the distribution (whether the image comes from the real distribution or fake distribution), and another to perform the self-supervised pretext task. We chose to explore two

prominent categories of such tasks: rotational learning and contrastive learning. With this, we trained generative models that can produce sample images with the additional context of the pretext task (e.g. the rotational self-supervised GAN can generate rotated images). The work we have done here is a promising step in exploring methods of producing stable GANs without the cost of needing labelled data like in conditional GANs or computationally expensive operations like in various types of Wasserstein GANs.

2 Background

2.1 Generative Adversarial Networks

As discussed previously, GANs are generative models. They formulate the problem as a two player min-max game where one is a generator that tries to learn the distribution from the random latent space and the other is a discriminator that tries to classify whether the distribution comes from the real or fake data. Typically a generator is a feed forward network that draws the distribution from input noise $Z \sim \mathcal{N}(0, I)$ while the discriminator network is a binary classifier that outputs the probability of data produced by the generator being real or fake. Both these networks are simultaneously trained until convergence, with the discriminator being trained more times than the generator for each epoch as the feedback from the discriminator is generally more valuable for a successful training. The vanilla GAN framework with loss function shown can be seen in the Fig. 1.

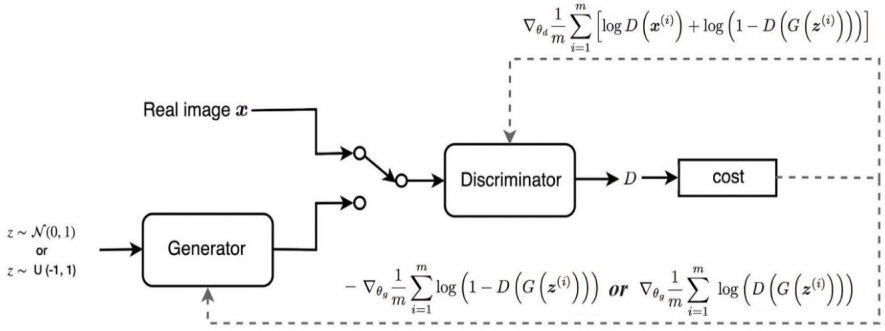


Figure 1: Vanilla GAN

As it could be seen, the goal of the discriminator is to maximize the loss $\mathbb{E}_{w \sim p_{\text{data}}}[\log D(w)] + \mathbb{E}_{z \sim \mathcal{N}(0,I)}[\log(1 - D(G(z)))]$, whereas the generator aims to minimize the loss $\mathbb{E}_{z \sim \mathcal{N}(0,I)}[\log(1 - D(G(z)))]$.

2.2 Forgetful Discriminator

We have seen from the above description of loss functions that the learning of the discriminator depends heavily on the distribution generated by the generator. Since the constant changes in the parameters of the generator cause the distribution to change as well, there is a non-stationary learning environment for the discriminator. This means that the discriminator is prone to discriminate based on a current distribution, which is effective in classification, without accounting for earlier distributions, leading to a forgetful discriminator. This issue was addressed in Chen et al. [2019a] by introducing a self-supervised task, learning image rotations, to the discriminator and augmenting its loss function with the rotation-based loss. Our contribution is exploring contrastive learning for the self-supervised task and comparing its performance with rotation-based learning.

2.3 Self-Supervised Learning

Self-supervised learning, also known as pretext learning, is a representation learning method where the model trains itself to learn one part of the input from another part of the input. It is regarded as an intermediate between supervised and unsupervised learning because there is use of labels, but the labels are actually pseudo-labels attained by transformations on unlabeled data. The model is trained on a pretext task with generated pseudo-labels to learn useful feature representations, and this trained

model can be used to augment a downstream task by transfer learning. Fig. 3 shows a general SSL pipeline and Fig. 2 illustrates how rotational self supervised learning works.

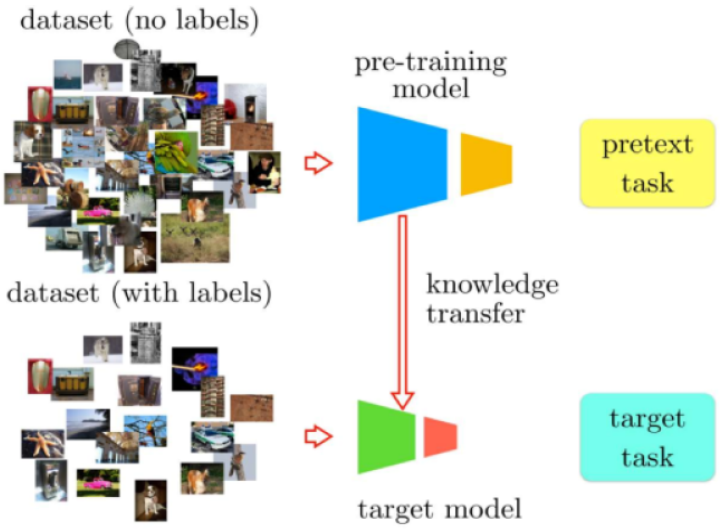


Figure 2: Illustration of self-supervised learning

We explore two self-supervised learning methods in this project: contrastive and rotational learning. Contrastive learning is an idea in representational learning where we try to minimize the distance between positive or similar samples and maximize the distance between negative samples. We measure the similarity between the samples using cosine similarity. In rotational learning, we are trying to learn feature representations that reflects the degree of rotation of the image. We induce these concepts in the discriminator to increase the learning stability and compare which self-supervised learning method works better in the image domain.

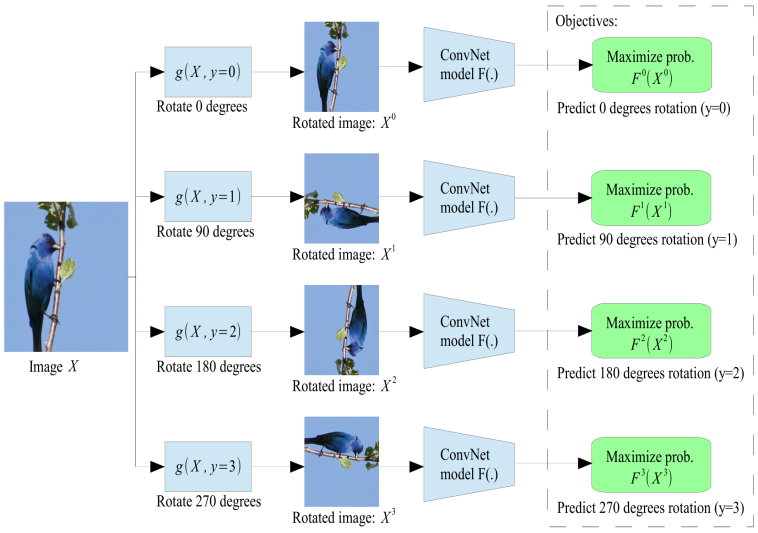


Figure 3: Illustration of Rotation Learning proposed by Gidaris et al. [2018]

3 Related Research

With the inception of GANs Goodfellow et al. [2014], performance and applications of generative modelling have evolved exponentially. Today, GANs are used in multiple domains such as image-to-image translation Zhu et al. [2017], domain adaptation Hu et al. [2018], and zero shot learning

Wang and Jiang [2019]. In particular, GANs are recently gaining recognition for their applications in self-supervised learning, the machine learning paradigm discussed in detail in the previous section. For example, Pathak et al. [2016] implemented an autoencoder with a GAN based solution to fill in the missing patches of the images, which served as an effective pretext task for subsequent tasks in classification, detection, and segmentation. We were particularly interested in applications where self-supervised methods were used to train more effective GANs: Chen et al. [2019a] uses a combination of self-supervision and adversarial training to bridge the gap between conditional and unconditional generative modelling, and Hou et al. [2021] proposes a new self-supervised GAN that addresses issues in previous works by introducing a label-augmented discriminator. To add to that, there has been work on inducing of rotational self-supervision Gidaris et al. [2018] in the discriminator model. Similar work is carried out by Chen et al. [2019b] where they use auxiliary rotation loss to embed the representation in the discriminator and ensure sparsity. Our idea takes inspiration to develop better methods for self-supervised learning.

4 Methods

In this section we intend to cover some important information regarding the experiment conducted in accordance for this project. It delineates in detail the model architecture, evaluation of models, and dataset. Also, the outline of the research pipeline is discussed for the formulation of optimization steps.

4.1 Dataset

Since we intend to perform generative modeling, which is an unsupervised task, we only need raw images for the task. To test out our hypothesis, we selected the anime face dataset Churchill and Chao [2019], which consists of around 143k anime face dataset drawn from different classes of images. Due to limitations of training resources and computational power, we decided to select a subset of these images: only 40k images were random selected with equal probability, and the models were trained on them. To keep the apple to apple comparison, we used the same subset across all the training strategies to achieve this task.

The main intent of our project is to check how well the models perform after inducing some self supervision in the discriminator. Since we are dealing with two different self supervision techniques, we needed transformations that could capture the underlying nuances. Rotation based self-supervised learning was easier since we just had to randomly rotate the images for the $angle \in [0°, 90°, 180°, 270°]$, which could be generated on the fly. To induce enough randomness, we were generating new rotations for the images each time so that the model did not simply learn from fixed rotations. For contrastive learning we used two different augmentation methods in series: adding random color jitter to distort the color channels of the image followed by the random crop (i.e crop the image randomly to capture every single patch or detail of the image rather than just simply cropping the image from center which only focuses on the central context of the image). All of the tricks mentioned above were used in preparing the dataset for the GANs.

4.2 Models

Just like any other traditional GAN model, our approach contains two different networks, the Generator (G) and the Discriminator (D). The generator aims to generate fake samples from the latent space, and the discriminator aims to detect whether images come from the real or fake distribution. We have modified the discriminator to address the issue at hand. Fig. 4 presents an outline of the model used in training the self supervised GANs. The discriminator is not a sequential model but is instead broken down into several small components with the back bone being the feature learner, which leverages a CNN to learn the high level as well low level features from the images. This feature learner is followed by the two MLP heads which are as follows:

- **Discriminative Head**: Just like any other discriminative head, this component is used to classify whether the image or sample comes from fake distribution or real distribution.
- **Self-Supervised Head**: This is the place where magic happens. This head spits out the output for the self supervised learning. In our case, it is either a classifier for classifying the

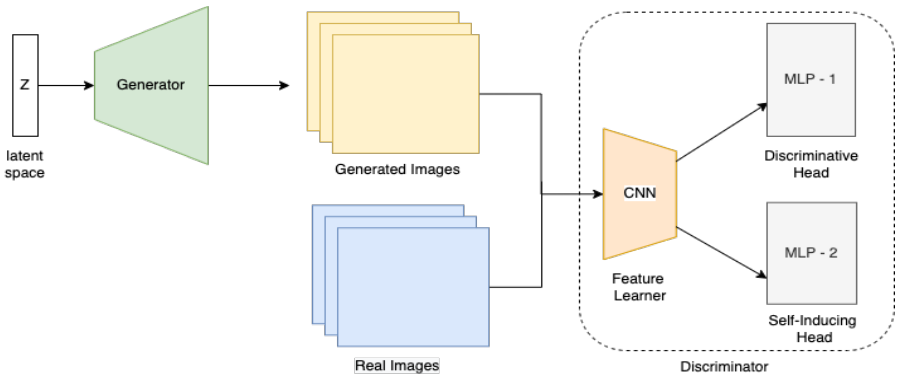


Figure 4: Overview of model architecture

orientation of the image or the encoder that aims to encode the feature representation for the contrastive learning. In the case of the vanilla GAN, this head could be pruned.

Generally, contrastive learning uses a projection head to compute cosine similarity between different samples with the aim of pulling similar features together and pushing distinct features apart. Therefore, while training the contrastive learning based GAN, we also include a projection head implemented as an MLP with an output of 128 latent dimensions.

4.3 Approach

We are using three different model architectures to train the GANs, but the basic underlying principles for training the self-supervised models remain the same as for the vanilla GAN where we are trying to achieve Nash equilibrium between the discriminator and the generator. Our addition is augmenting the discriminator to also consider the loss from the self-supervised learning component. The workflow for the generator is the same across all the models: a latent sample z is taken in and used to generate fake samples such that $\hat{x} = G(z)$. But in the case of the discriminator, it changes a little bit. First, we compute the output from the discriminator, $h_d = D(F(x))$, alongside the output for the self-supervised learning, $h_s = S(F(x))$, where F is the feature learner, D is discriminator head and S is the self-supervised learning head. This entire computation is done in a single pass to preserve the learning in the feature learner. Now in order to compute the loss for the generator, we also need to compute the discriminative output for the generated samples, which is $h_f = D(F(\hat{x}))$. Since we are only interested in checking how well our generator is performing, only the discriminative head in the discriminator is activated for this part.

Unlike in any other deep learning problem, we need to optimize all components of our network, so we use a stochastic gradient descent approach with Adam for moment based gradient optimization. Since we are using gradient descent, we need to define a loss function. Our modified loss functions are similar to those of a vanilla GAN with just an addition of the self supervised loss to the discriminator. Discriminator loss could be defined as $Loss_D = L_D(h_d, h_f) + SSL(h_s)$ where L_D corresponds to vanilla GAN loss for the discriminator and SSL refers to self supervised loss. The loss function for the generator, L_G, is unchanged from the vanilla GAN.

To understand the process better, consider the case of Rotnet based self-supervised learning. Here, the generator uses the given latent space to produces some samples, which are later passed down to the discriminator to compute the loss. For the images that come from the real distribution, we first augment them using the rotational transformation, giving us the images with some random orientation. These augmented images are then fed into the discriminator where they pass through the feature learner and then go into two different heads MLP_D & MLP_S, the discriminative head and self-supervised head respectively. In the discriminative head, we compute the standard discriminator loss, and in the self-supervised head, we solve the classification problem for determining the orientation of the image. During the optimization step both losses are combined as a whole, and discriminator tries to minimize this aggregate. Things get somewhat convoluted when we shift gears to contrastive learning. The procedure for the fake samples is the same as defined above for the Rotnet, but it becomes interesting for the real distribution. In the augmentation step, we retain both the original

images and their augmented versions. For the augmented images, we are interested in computing the contrastive loss, so only MLP_S is activated, and its output is sent to the projection head, which regularizes the results. For the original images, we get outputs from both MLP_S&MLP_D. MLP_D focuses on the discriminative loss, and MLP_S tries to minimize the self-supervised learning loss. Output from MLP_S is again fed to the projection head to carry out the task of minimizing the contrastive loss. This is the detailed run-through of our method.

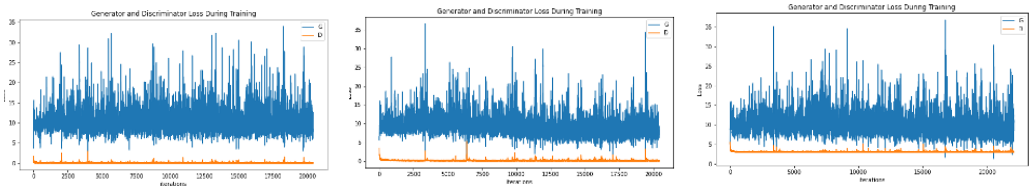
The way we perform optimization could be done using an alternative method: instead of training both self-supervised and discriminative heads in a single pass, we can first train the discriminator on a pretext task, just like in DCGAN Radford et al. [2015] where they used a pre-trained discriminator on ImageNet. It would be interesting to experiment on this method with other representation learning tasks, similar to the one described by Mishra et al. [2021] where they pre-train their models on a different representation learning task and later use it to train the model for domain adaptation in semi-supervised learning. In our case, we can replace domain adaptation with generative adversarial training.

4.4 Evaluation

In order to perform all the experiments in an unbiased fashion, we kept all hyperparameters for the shared parts of the three models the same (i.e. all parts besides the self supervised head). For all three models, we used the entire subset of data randomly drawn from the main data source, keeping a batch size of 64. Since we do not have any definite method to measure the performance of generated samples, we had to stick with the standard method of comparison for measuring distance between two image distribution. There are generally two such methods available, one being Inception Score (IS) and another being Frechet Inception Distance (FID). For our experiment, we decided to use the latter. To generate FID based scores for the generator, we needed to perform the following steps: first pass the real distribution to the *InceptionV3* model, which generates a feature space of 2048 dimensions. Repeat the same process for the images generated by the generator. Once both the vectors have been computed, we can calculate the FID based score using eq. 1

$$\text{FID} = \|\mu - \mu_w\|_2^2 + \text{tr}\left(\Sigma + \Sigma_w - 2\left(\Sigma^{1/2}\Sigma_w\Sigma^{1/2}\right)^{1/2}\right) \tag{1}$$

5 Results



(a) Vanilla GAN (b) Rotation Net GAN (c) Contrastive Learning GAN

Figure 5: Training loss of the GANs

The three different types of GANs that we have discussed were trained and tested on the data set collected by Churchill and Chao [2019]. Table 1 summarizes the performance of the GANs, measured by their ability to accurately reproduce the images from the dataset. As previously mentioned, we used the FID based score, where a lower value is better. Fig. 5 shows the loss curves for the three types of GANs trained. Due to limitations in computational resources, we were only able to train the models for $21,000$ iterations, which is generally a low amount for this domain. As a result, we were not able to get great FID based scores, and our generated images were far away from the real distribution. However, we can see that the loss curve seemingly shows improvement over the iterations.

After training the models, we used each of them to generate image samples. Fig. 6a shows the samples generated by the vanilla GAN model. From our limited samples, we can observe that a high number of images had been distorted. The results for the rotnet based GAN show some improvement, as seen in Fig. 6b, though these improvements are rather minor compared to the vanilla GAN. We

(a) Vanilla GAN (b) Rotation Net GAN (c) Contrastive Learning GAN

Figure 6: Image samples produced from the GANs

are able observe the rotation being done by the rotnet GAN on the samples it generates. The sample images generated by the contrastive learning GAN in 6c are much less laden with distortions.

Table 1: FID based scores for the different types of GANs

Vanilla GAN	RotationNet GAN	Constrastive Learning GAN
228.58897	223.26015	204.56511

6 Conclusion

In order to improve the stability of GANs, we thought to add a component of self-supervised learning to the discriminator. Both models that we explored outperformed the vanilla GAN as measured by the FID based scores on the generators. We also qualitatively observed less distortions in the generated images of the self-supervised GANs. In particular, the Contrastive Learning GAN showed great improvements. Overall, although we did not directly compare our models to alternative variations of GANs that seek to solve similar issues, we see promise in using self-supervised GANs to learn models that produce high quality samples. Our experiments show that using self-supervised learning based approaches successfully mitigates the forgetful discriminator and helps improve the performance of GANs.

7 Future Work

In terms of fine-tuning the models, we can extend our ideas by applying other forms of self-supervised learning, such as context encoders in search for better performance. Potentially, the self-supervised approach in GANs could also be applied to semi-supervised models.

Although training with additional resources could alleviate the issue of FID based scores being higher than usual, another solution is to employ another metric. Based on the idea that GANs attempt to achieve Nash equilibrium, we could use the measure of how close the value of D is to the optimal discriminator at the Nash equilibrium, which should ideally be $D(x) = \frac{1}{2}$.

There is room for us to explore more precisely how self-supervised learning helps in learning the model. One idea is to try regularizing the self-supervised learning loss with discriminator loss and checking how much it impacts the training of the models.

Finally, we can explore how well our models deal with other issues such as mode collapse, and whether our methods can be combined with other variations of GANs to overcome these issues.

Collaboration

This project is a collaborative effort between all the authors. As such, it is important to define the responsibilities upfront to avoid redundancies and conflicts. Based on mutual understanding and agreement, we devised the following plan of action:

- Data Preprocessing and Loading Pipeline: Rishav Sen
- Training of VanillaGAN: Daniel Shu
- Training of RotNetGAN: Jatin Kodali
- Training of Contrastive Learning: Sanidhya Mangal & Rishav Sen
- Presentation and Documentation: All Authors

Additional Resources

The code base for the entire experiment can be found at this link. All the instructions for running the code and downloading the dataset are present in the readme file.

References

Ting Chen, Xiaohua Zhai, Marvin Ritter, Mario Lucic, and Neil Houlsby. Self-supervised gans via auxiliary rotation loss. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019a.

Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised representation learning by predicting image rotations. *arXiv preprint arXiv:1803.07728*, 2018.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014.

Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2223–2232, 2017.

Lanqing Hu, Meina Kan, Shiguang Shan, and Xilin Chen. Duplex generative adversarial network for unsupervised domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1498–1507, 2018.

Jinghua Wang and Jianmin Jiang. Conditional coupled generative adversarial networks for zero-shot domain adaptation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3375–3384, 2019.

Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A. Efros. Context encoders: Feature learning by inpainting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

Liang Hou, Huawei Shen, Qi Cao, and Xueqi Cheng. Self-supervised gans with label augmentation. *arXiv preprint arXiv:2106.08601*, 2021.

Ting Chen, Xiaohua Zhai, Marvin Ritter, Mario Lucic, and Neil Houlsby. Self-supervised gans via auxiliary rotation loss. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12154–12163, 2019b.

Spencer Churchill and Brian Chao. Anime face dataset, 2019. URL `https://www.kaggle.com/ds/379764`.

Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.

Samarth Mishra, Kate Saenko, and Venkatesh Saligrama. Surprisingly simple semi-supervised domain adaptation with pretraining and consistency. *arXiv preprint arXiv:2101.12727*, 2021.