DSA5204 Final Report

Semester 2 AY 21/22

Generative Adversarial Active Learning

Type of project: Algorithm Development

Zhao Yuxiong (A0236001B), Shaunn Tan De Hui (A0087785H)
Li Zitian (A0235944X), Zhang Shaoxuan (A0080411X)
Ong Jian Ying Gary (A0155664X), Haoyang Liu (A0235954W)

Abstract

Labelling of data for supervised training is time consuming and costly. The authors in the selected paper proposed a novel training framework, Generative Adversarial Active Learning (GAAL), that combines Active Learning and Generative Adversarial Networks. GAAL attempts to utilize informative synthetic data generated from a GAN to increase the training speed of a learner, a Support Vector Classifier (SVC) in the authors' work. We replicated the main results from the authors' experiments, compared our results and proposed two extensions, replacing the Deep Convolutional-GAN with a Wasserstein GAN and incorporating a diversity measure in the objective function, in an attempt to improve the accuracy of the SVC.

1 Introduction

The paper selected for this project is titled "Generative Adversarial Active Learning" by Zhu and Bento (2017). Detailed exposition of the discussed methods and models are not included in this report to keep it concise. We refer the reader to the cited articles for the full details of each method.

1.1 Active Learning

Compared to labelled data, unlabelled data is relatively abundant. However, in order to be able to perform a supervised learning task, these unlabelled data will need to be labelled and the process is time consuming and costly.

Active learning algorithms seek to maximize the accuracy of trained learners with fewer labelled training samples through strategic selection of samples from a pool of unlabelled data for querying and labelling, then adding these samples to a labelled pool to update the learner.

Such strategic queries can be from a given unlabelled pool such as in Active SVM (Tong and Chang (2001)), generated from the some distributions (Atlas, Cohn, and Ladner (1989)), or sampled randomly from the unlabelled pool. Some common metrics used are distance (Tong and Chang (2001)) and uncertainty (Lewis and Gale (1994)). A thorough review on active learning approaches can be found in Settles (2009).

1.2 Generative Adversarial Network

Generative Adversarial Networks (GANs) have received much attention since the work by Goodfellow et al. (2014). A GAN is commonly described as a two-player minimax game between a generator G and a discriminator D:

$$\min_{\theta_1} \max_{\theta_2} \left[\mathbf{E}_{x \sim p_{data}} \log D_{\theta_1}(x) + \mathbf{E}_z \log(1 - D_{\theta_1}(G_{\theta_2}(z)))\right] \tag{1}$$

where p_{data} is the underlying distribution of the real data and z is some uniformly distributed random variable. The basic algorithm of a GAN is depicted in Figure 1. We train a generator and discriminator by performing updates to the generator such that it eventually generates synthetic data which the discriminator is not able to determine if it is real or fake.

GANs have since been widely applied in areas such as image processing (Li and Wand (2016), Tran, Yin, and Liu (2017), and Vondrick, Pirsiavash, and Torralba (2016)) and natural language processing (Qiao et al. (2019) and Lin et al. (2017)). There are also literature that deliver more detailed reviews on the algorithm and application (Gui et al. (2020), Aggarwal, Mittal, and Battineni (2021), and Yi, Walia, and Babyn (2019)).

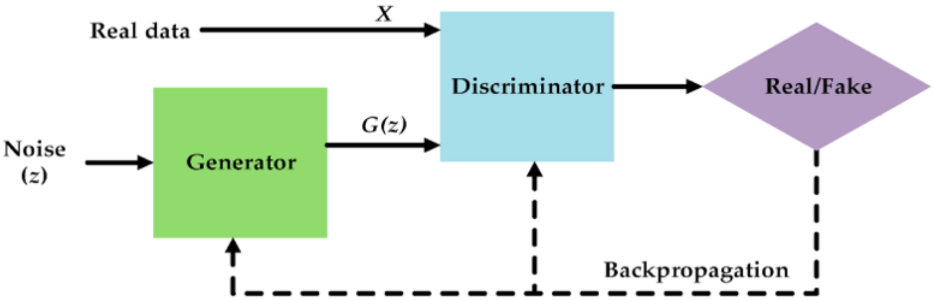


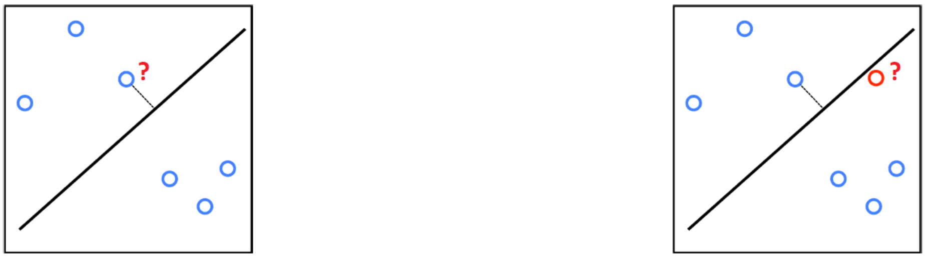
Figure 1: Generative Adversarial Network (GAN)

1.3 Generative Adversarial Active Learning

The selected paper introduces a novel active learning framework by introducing a query synthesis approach that combines aspects of Active Learning and GANs, aptly named "Generative Adversarial Active Learning". Firstly, utilising a generator from a GAN trained on the pool of unlabelled data, synthesize informative training samples that are adapted to the learner that is being trained. Then, human oracles label these synthetic samples and add them to the labelled pool to update the learner. Iterate these steps till the labelling budget is reached.

In the selected paper, the authors specified the learner to be trained as a Support Vector Classifier (SVC) and the active learning synthesis problem is defined as:

$$\min_{z} \| W^{\intercal}\phi(G(z)) + b \| \tag{2}$$



(a) Active SVM: The sample closest to the decision boundary is queried.

(b) GAAL: A synthetic sample close to the decision boundary is generated and queried.

Figure 2: Compare querying methods of Active SVM and GAAL

In Figure 2, we compare the pool-based approach in Active SVM which selects unlabelled samples closest to the decision boundary for querying (as in (a)) and the GAAL problem (as in Equation 2) which generates new synthetic samples closest to the boundary for querying (as in (b)). The GAAL algorithm is as follows:

Algorithm 1 Generative Adversarial Active Learning (GAAL)

```
Require: (1) A generator $G$ trained on all unlabelled data, (2) Labelled training dataset $S$ initialized by randomly
    picking a small fraction of the unlabelled data to label, (3) $J$ = number of new synthetic samples per batch
while labelling budget not reached do
    for $j = 1$ to $J$ do
        Solve the optimization problem in equation (2) by descending the gradient:
                    $\nabla_z \|W^{\intercal}\phi(G(z)) + b\|$        (3)
        Use the solutions $\{z_1, \dots, z_J\}$ and $G$ to generate instances for querying
        Label $\{G(z_1), \dots, G(z_J)\}$ by human oracles
        Add the labelled data to the training dataset $S$
    end for
    Update the learner, $W$ and $b$
end while
```

2 Replication & Experiments

2.1 Replication

The GAAL algorithm presented in the selected paper, prescribes the use of a human oracle to label synthetic images. Our replication utilises a pre-trained convolutional neural network (VGG-16) as the human oracle.

We attempted the replication of the experiments conducted by the authors, comparing the performance of the GAAL algorithm with 4 other training schemes:

Training Scheme	Description
GAAL	As described in Algorithm 1
Simple GAN	Generate new training data with G, label and add to training dataset after oracle labels them
Active SVM	Select instances from the unlabelled pool that is closest to the boundary, label and add to training dataset
Random Sampling	Randomly select instances from the unlabelled pool, label and add to training dataset
Fully Supervised	Using all samples in the labelled pool to train the classifier

The experiments were done with the following dataset combinations:

Training Set	Testing Set
MNIST 5 & 7	USPS 5 & 7
MNIST 5 & 7	MNIST 5 & 7
CIFAR-10 Horse & Automobile	CIFAR-10 Horse & Automobile

There are some differences in results from our replication when compared to the authors' experiments. In the following charts, the results from the authors are presented on the left while those from our replication are presented on the right:

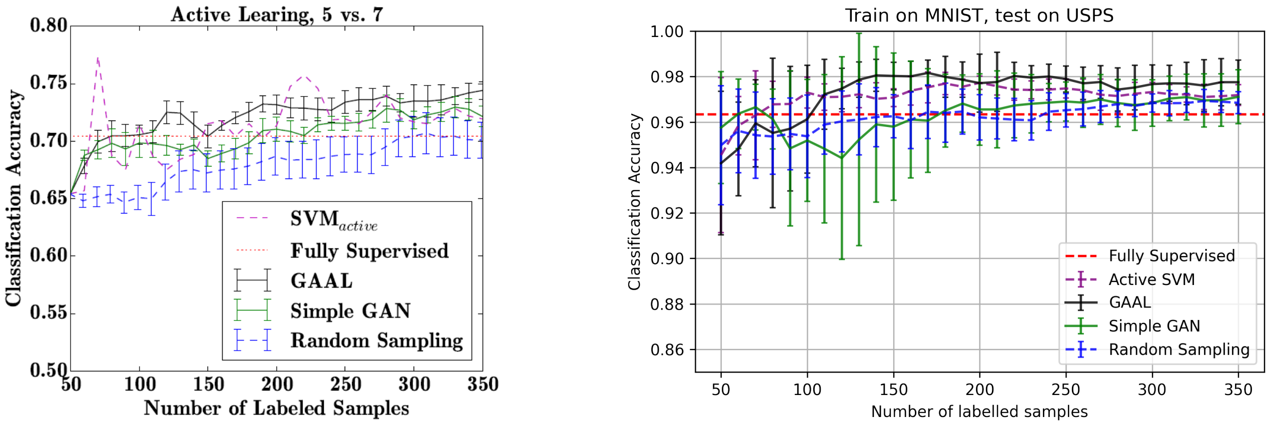


Figure 3: Train on MNIST, test on USPS

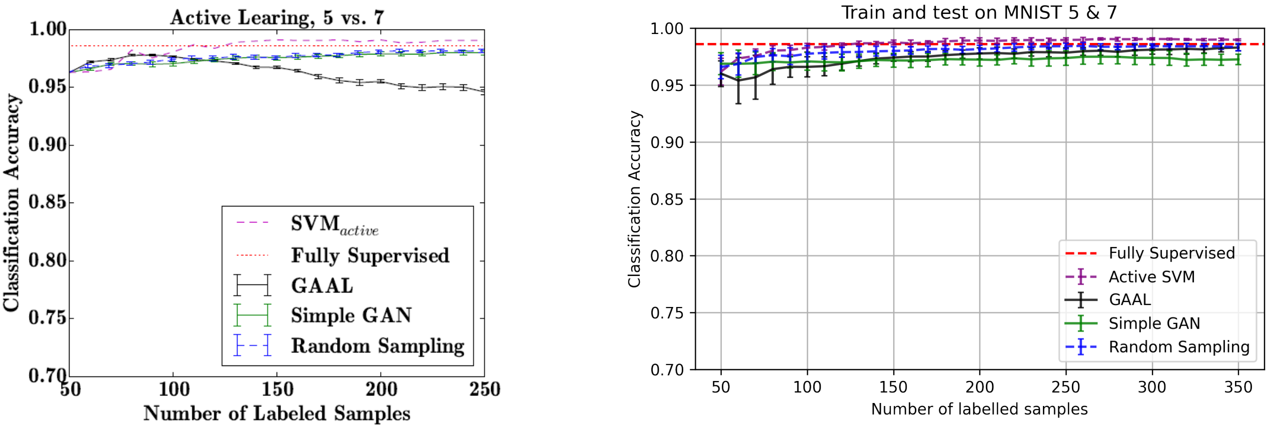


Figure 4: Train and test on MNIST

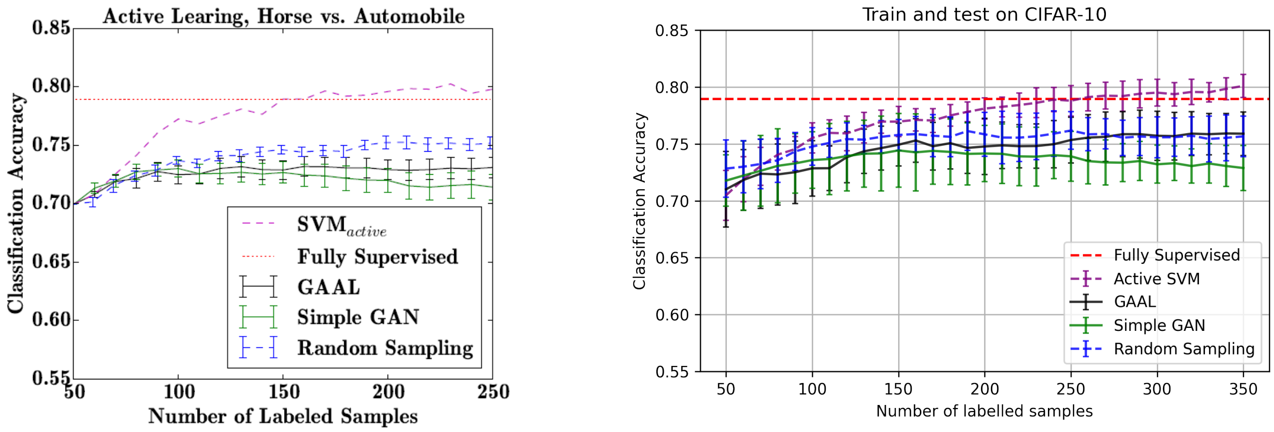


Figure 5: Train and test on CIFAR-10

2.2 Difference in Replication Results

We noted two key observations:

1. Our replication had a substantially higher test accuracy for the USPS dataset, and,
2. Our SVC accuracy (relative to authors' results) is higher under GAAL training for all three experiments.

2.2.1 Higher USPS Test Accuracy

Referencing Figure 3, we observed a higher test accuracy in our replication with mean of 98% for GAAL training at 350 samples, as opposed to approximately 73% from the authors' results.

The authors' preprocessing methods for the USPS test data was not specified, as such, we hypothesise that the higher test accuracy is due to different preprocessing of the USPS test dataset in our replication. We added some padding to the USPS image samples during upsampling from 16x16 to 28x28 in order to replicate the structure of MNIST samples. An example of our processed USPS test data is shown in the third picture in Figure 6:

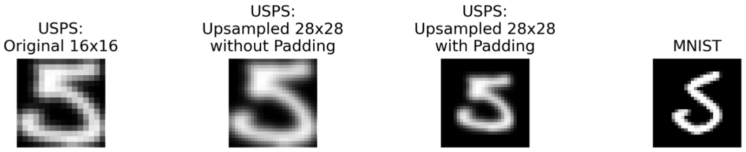


Figure 6: An example of our processed USPS dataset in the third picture

2.2.2 Higher GAAL Test Accuracy

Intuitively, with GAAL training relying on synthetic data to train the SVC, a poorly trained generator that is not able to capture the distribution of the unlabelled data will not be able to generate representative synthetic samples for labelling:

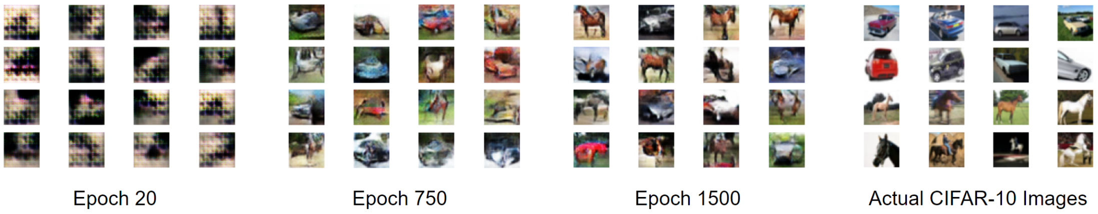


Figure 7: Quality of synthetic images improves when generator is updated for more epochs

From Figure 7, clearly, performing GAAL training with an Epoch 20 generator will perform poorly due to the poor quality of the synthetic images.

We attempted GAAL training using generators updated for different number of epochs[1]. Interestingly, the algorithm performed best with the generator updated for 1000 epochs (see Figure 8).

[1] We define an epoch as having updated the discriminator and generator (*No. of training samples in pool/Batch size*) times

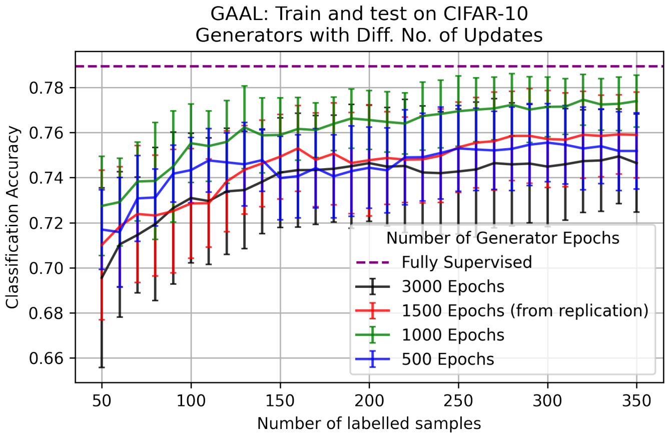


Figure 8: Comparing test accuracy of learner under GAAL with different generators

3 Extensions

3.1 Wasserstein GAN & More Generator Channels to Improve Synthetic Images

GAAL training is sensitive to the quality of the synthetic data from the generator. We attempted to increase the accuracy of the SVC, by utlilizing a Wasserstein GAN (WGAN) in place of the DC-GAN in view of some of its benefits (Arjovsky, Chintala, and Bottou (2017)): (1) meaningful loss metric, (2) stability of optimisation process and (3) no mode collapse.

Additionally, we also experimented with doubling the number of channels of the last Conv2DTranpose layer before the final Conv2D layer in an attempt to increase the quality of synthetic images. The results are presented in the following charts:

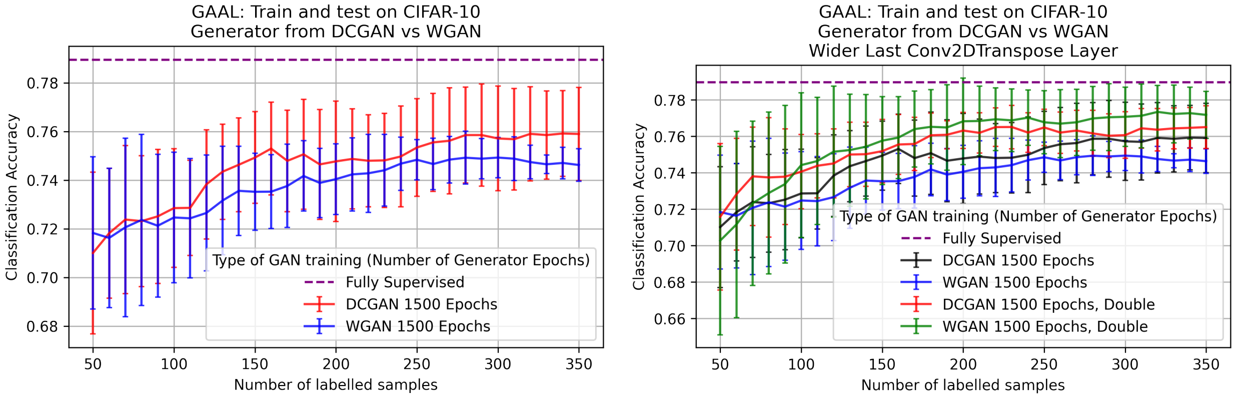


Figure 9: Results from increasing the quality of synthetic images

Figure 9 (left) shows that utilising a generator from a WGAN does not significantly increase the SVC accuracy (note the overlapping 1-SD error bars). Increasing the size of the Generator used for GAAL training increases the mean accuracy of the classifier (Figure 9 (right)), but likewise, the increase appears to be insignificant. It is worth exploring alternative GAN architectures to improve the generator used for GAAL.

3.2 Adding Diversity Measure (Average Distance) in Objective Function

Cai et al. (2021) describes a Max-Min Greedy Active Selection algorithm that attempts to maximize both the minimum uncertainty and minimum distance between selected samples when selecting samples from the candidate pool to add to the labelled pool. Following the same intuition, in order to synthesise samples that are both informative (as characterized by uncertainty) and diverse, we can add a similar diversity measure to the objective function defined in Equation 2, penalizing small average distance between the synthetic sample $G(z)$ and all other samples x_i in the labelled pool $\mathcal{L}$. Consider a modified optimization problem from Equation 3 in Algorithm 1:

$$\nabla_z \|W^\intercal \phi(G(z)) + b\| - \lambda \frac{1}{N} \sum_{i=1}^{N} \|G(z) - x_i\| \tag{4}$$

where $i \in \{1, \ldots, N\}$, N is the number of instances in labelled pool $\mathcal{L}$ and λ is a hyperparameter for diversity penalty.

We repeated the GAAL experiment on the CIFAR-10 dataset, with the additional diversity measure and $\lambda \in \{0.0001, 0.001, 0.01, 0.1, 1\}$. With reference to our GAAL training result from Figure 5 (No Penalty), we compare the new results incorporating diversity penalty in the following chart:

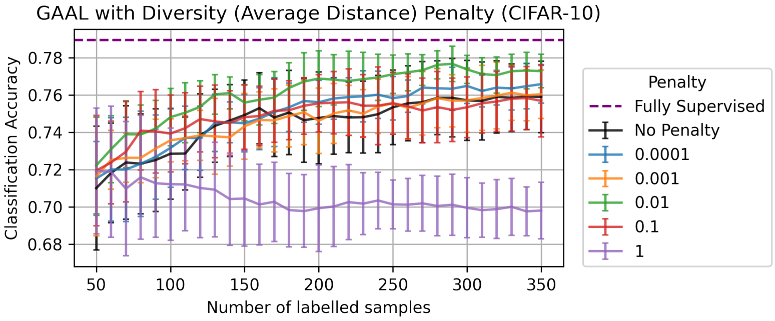


Figure 10: Results from including diversity measure

From Figure 10, we noted that adding a diversity penalty to the optimisation problem does indeed affect the rate of increase of accuracy of the SVC which achieved a higher average accuracy at 350 labelled samples when $\lambda = 0.01$, while $\lambda = 1$ is clearly detrimental to ability of the SVC to learn from the synthetic images.

4 Summary & Further Work

In this report, we provided a summary of GAAL and presented the results from our replication of the experiments by the authors. We noted some differences in our replication results and provided some insights to the likely causes of the differences. We proposed two extensions to the authors' work: (1) replacing the DC-GAN with a WGAN in view of some of its benefits and changing the size of the generator, and (2) adding a diversity measure to the GAAL objective function.

We noted no clear benefits from using a Generator from a WGAN, but as expected, increasing the size of the Generator has a positive impact on the accuracy of the SVC. Adding a diversity measure also had an impact on the accuracy of the SVC, and its impact is affected by the tunable hyperparameter. We observed best accuracy with $\lambda = 0.01$, while utilizing $\lambda = 1$ led to poorer accuracy of the model.

From our observations, the GAAL training algorithm is sensitive to the quality of the Generator. Additional work into improving the Generator is likely able to improve the learner's accuracy. Further, a convolutional neural network could be used in place of the SVC.

Noting that diversity measures appear to have an impact on the synthetic images, and consequently the accuracy of the learner, we could consider utilising other varieties of measures, such as cosine similarity, in the objective function. Also, the penalty factor is tunable and could be fine-tuned to improve the accuracy of the learner when its trained under GAAL.

5 Contributions

Name	Student ID	Contribution
Zhao Yuxiong	A0236001B	Train a DC-GAN, replicate GAAL experiment with MNIST 5 & 7, extended with Wasserstein GAN
Shaunn Tan De Hui	A0087785H	Train a DC-GAN, replicate GAAL experiment with MNIST 5 & 7, extended with Wasserstein GAN
Li Zitian	A0235944X	Train a DC-GAN, replicate GAAL experiment with MNIST 5 & 7, extended with Wasserstein GAN
Zhang Shaoxuan	A0080411X	Replicate Active SVM and GAAL experiment with CIFAR-10, extended with diversity measure
Ong Jian Ying Gary	A0155664X	Replicate Active SVM and GAAL experiment with CIFAR-10, extended with diversity measure
Haoyang Liu	A0235954W	Replicate Active SVM and GAAL experiment with CIFAR-10, extended with diversity measure

References

Aggarwal, Alankrita, Mamta Mittal, and Gopi Battineni (2021). “Generative adversarial network: An overview of theory and applications”. In: *International Journal of Information Management Data Insights* 1.1, p. 100004. ISSN: 2667-0968.

Arjovsky, Martin, Soumith Chintala, and Léon Bottou (2017). “Wasserstein GAN”. In: DOI: 10.48550/ARXIV.1701.07875.

Atlas, Les, David Cohn, and Richard Ladner (1989). “Training Connectionist Networks with Queries and Selective Sampling”. In: *Advances in Neural Information Processing Systems*. Vol. 2. Morgan-Kaufmann.

Cai, Lile et al. (2021). “Exploring Spatial Diversity for Region-Based Active Learning”. In: *IEEE Transactions on Image Processing* 30, pp. 8702–8712. DOI: 10.1109/TIP.2021.3120041.

Goodfellow, Ian et al. (2014). “Generative Adversarial Nets”. In: *Advances in Neural Information Processing Systems*. Vol. 27. Curran Associates, Inc.

Gui, Jie et al. (2020). “A Review on Generative Adversarial Networks: Algorithms, Theory, and Applications”. In: *arXiv:2001.06937 [cs, stat]*. arXiv: 2001.06937.

Lewis, David D. and William A. Gale (1994). “A Sequential Algorithm for Training Text Classifiers”. In: *SIGIR '94*. Ed. by Bruce W. Croft and C. J. van Rijsbergen. London: Springer, pp. 3–12. ISBN: 978-1-4471-2099-5.

Li, Chuan and Michael Wand (2016). “Precomputed Real-Time Texture Synthesis with Markovian Generative Adversarial Networks”. In: *Computer Vision – ECCV 2016*. Ed. by Bastian Leibe et al. Cham: Springer International Publishing, pp. 702–716. ISBN: 978-3-319-46487-9.

Lin, Kevin et al. (2017). “Adversarial Ranking for Language Generation”. In: vol. 30. Curran Associates, Inc.

Qiao, Tingting et al. (2019). “MirrorGAN: Learning Text-To-Image Generation by Redescription”. In: *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). Long Beach, CA, USA: IEEE, pp. 1505–1514. ISBN: 978-1-72813-293-8.

Settles, Burr (2009). *Active Learning Literature Survey*. Computer Sciences Technical Report 1648. University of Wisconsin–Madison.

Tong, Simon and Edward Chang (2001). “Support vector machine active learning for image retrieval”. In: *Proceedings of the ninth ACM international conference on Multimedia*. MULTIMEDIA '01. New York, NY, USA: Association for Computing Machinery, pp. 107–118. ISBN: 978-1-58113-394-3.

Tran, Luan, Xi Yin, and Xiaoming Liu (May 31, 2017). “Representation Learning by Rotating Your Faces”. In: *https://arxiv.org/abs/1705.11136* PP.

Vondrick, Carl, Hamed Pirsiavash, and Antonio Torralba (2016). “Generating Videos with Scene Dynamics”. In: *Advances in Neural Information Processing Systems*. Vol. 29. Curran Associates, Inc.

Yi, Xin, Ekta Walia, and Paul Babyn (2019). “Generative Adversarial Network in Medical Imaging: A Review”. In: *Medical Image Analysis* 58, p. 101552. ISSN: 13618415.

Zhu, Jia-Jie and José Bento (2017). “Generative Adversarial Active Learning”. In: *CoRR* abs/1702.07956. arXiv: 1702.07956.