CLIP-MDGAN: Multi-Discriminator GAN Using CLIP Task Allocation

Shonosuke Gonda, Fumihiko Sakaue and Jun Sato
Nagoya Institute of Technology, Nagoya 466-8555, Japan
gonda@cv.nitech.ac.jp, {sakaue, junsato}@nitech.ac.jp

Keywords: Image Synthesis, Image Distribution, GAN, Multi-Discriminator, Clip, Foundation Model, Multimodal.

Abstract: In a Generative Adversarial Network (GAN), in which the generator and discriminator learn adversarially, the performance of the generator can be improved by improving the discriminator's discriminatory ability. Thus, in this paper, we propose a method to improve the generator's generative ability by adversarially training a single generator with multiple discriminators, each with different expertise. By each discriminator having different expertise, the overall discriminatory ability of the discriminator is improved, which improves the generator's performance. However, it is not easy to give multiple discriminators independent expertise. To address this, we propose CLIP-MDGAN, which leverages CLIP, a large-scale learning model that has recently attracted a lot of attention, to classify a dataset into multiple classes with different visual features. Based on CLIP-based classification, each discriminator is assigned a specific subset of images to promote the development of independent expertise. Furthermore, we introduce a method to gradually increase the number of discriminators in adversarial training to reduce instability in training multiple discriminators and reduce training costs.

1 INTRODUCTION

In recent years, advances in research in VAE (Variational Auto Encoder) (Kingma and Welling, 2014), GAN (Generative Adversarial Network) (Goodfellow et al., 2014), and diffusion models (Ramesh et al., 2021; Rombach et al., 2022) have made it possible to reproduce the image distribution within datasets accurately and to generate high-quality images. Among these approaches, GANs have demonstrated applications not only in high-fidelity image generation (Brock et al., 2019) (Karras et al., 2020) but also in diverse tasks such as domain translation (Karras et al., 2020), super-resolution (Ledig et al., 2017), text-to-image generation (Zhang et al., 2017), and anomaly detection (Zhang et al., 2017). The mechanism of GAN, which improves capabilities by competing among multiple networks, is very important in advancing learning methodologies.

In GANs, training typically progresses by pairing one Generator with one Discriminator, allowing them to learn in competition. As the Discriminator's ability to distinguish improves, so does the Generator's ability to produce realistic images. If we think of the Generator as a student and the Discriminator as a supervisor, students will be able to acquire a broader and higher level of ability if they are taught by multiple supervisors, each with different specialized expertise, rather than by a single supervisor. Thus, we in this paper propose a method to enhance the Generator's capabilities by adversarially training it with multiple Discriminators, each with different expertise, as shown in Figure 1.

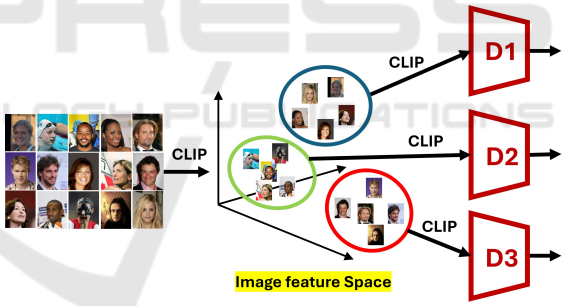


Figure 1: A method for automatically selecting the Discriminator corresponding to each training image based on image feature distribution.

Several methods have previously been proposed to enhance Generator performance using multiple Discriminators (Dinh Nguyen et al., 2017; Durugkar et al., 2017; Choi and Han, 2022). However, task allocation to multiple Discriminators is not a trivial problem and it often suffers from instability. Thus, in this paper, we propose a method for allocating tasks to multiple Discriminators by using CLIP (Contrastive

Gonda, S., Sakaue, F. and Sato, J.
CLIP-MDGAN: Multi-Discriminator GAN Using CLIP Task Allocation.
DOI: 10.5220/0013231900003912

In *Proceedings of the 20th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2025) - Volume 2: VISAPP*, pages 464-470
ISBN: 978-989-758-728-3; ISSN: 2184-4321

Language-Image Pre-training) (Radford et al., 2021), a large-scale model pretrained on the relationship between images and texts using large amount of image and text data.

In recent years, the advent of large-scale models pretrained on vast datasets has enabled zero-shot performance across various downstream tasks by utilizing well-trained Encoders within these large-scale models. In light of this, we propose an efficient and stable method for adversarial training between a Generator and multiple highly specialized Discriminators by leveraging the Encoder of CLIP to streamline the specialization of multiple Discriminators. We call it CLIP-MDGAN.

As illustrated in Figure 1, our CLIP-MDGAN selects the appropriate Discriminator for each image based on its image features. By strategically employing CLIP's Text Encoder and Image Encoder, we can cluster image groups effectively for each Discriminator. Since CLIP Encoders are trained to capture correlations between images and text, the feature vectors from its Image Encoder serve as valuable symbolic descriptors, facilitating meaningful clustering of large datasets. Hence, by clustering images into multiple classes using these feature vectors, we assign each Discriminator a specific classification task. This assignment allows each Discriminator to focus on identifying images with distinct characteristics, fostering high specialization among multiple Discriminators and enhancing the Generator's performance through adversarial training. Our approach avoids the instability seen in the existing methods during Discriminator selection, while enabling each Discriminator to develop high specialization efficiently.

This paper presents both a semi-automatic and a fully automatic approach for assigning tasks to multiple Discriminators using CLIP. Additionally, to ensure fast and stable task allocation to multiple Discriminators, we introduce a method that gradually increases the number of Discriminators in stages.

While this study focuses on enhancing a single Generator with multiple Discriminators, this approach can be extended to frameworks involving multiple Generators (Ghosh et al., 2018; Hoang et al., 2018), potentially enabling multi-Generator systems to benefit from adversarial training with specialized Discriminators.

2 RELATED WORK

Several methods have previously been proposed to enhance Generator performance through the use of multiple Discriminators. For example,

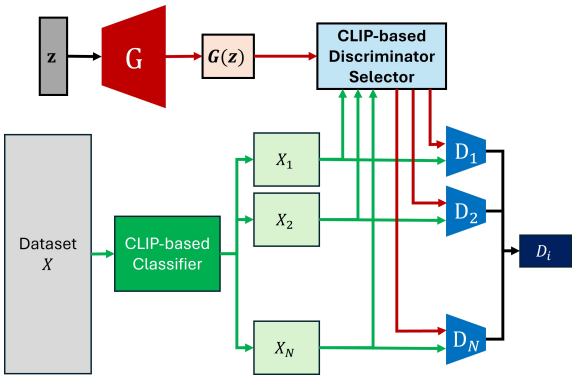


Figure 2: Overview of CLIP-MDGAN.

D2GAN (Dinh Nguyen et al., 2017) demonstrated that employing two Discriminators with opposing loss functions could help avoid mode collapse in GANs. Albuquerque (Albuquerque et al., 2019) introduced multi-objective discriminators, each focused on distinct tasks, thereby enhancing the Generator's performance. GMAN (Durugkar et al., 2017) proposed a balanced approach by combining a hard Discriminator (strict teacher) and a soft Discriminator (lenient teacher) to regulate the Generator's learning.

There have also been attempts to instill specialized skills within multiple Discriminators during training. MCL-GAN (Choi and Han, 2022) utilizes Multiple Choice Learning (MCL) (Guzman-Rivera et al., 2012) to select a limited number of Discriminators from a larger pool based on each dataset image, facilitating adversarial learning with the Generator. However, this method requires multiple Discriminators for the classification of a single data instance, resulting in overlapping roles among these Discriminators. Additionally, during the early stages of training, task allocation to each Discriminator can be unstable, making it challenging to maintain operational stability.

Thus, we in this paper propose a method for allocating tasks to multiple Discriminators by using CLIP (Radford et al., 2021), a large-scale model pretrained on the relationship between images and texts. By using the well-trained text and image encoders of CLIP, our method can avoid instability during Discriminator selection, while enabling each Discriminator to develop specialization efficiently.

3 TRAINING GENERATORS WITH MULTIPLE DISCRIMINATORS

The overall view of our CLIP-MDGAN is as shown in Figure 2.

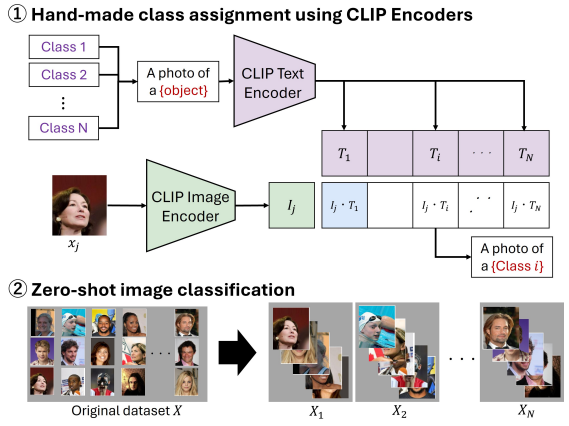


Figure 3: Hand-made class assignment for multiple discriminators.

First, we divide Dataset X into N classes X_i $(i = 1, \cdots, N)$ by using CLIP Encoders. Next, the CLIP-based Discriminator Selector identifies which of these N classes the image $G(z)$ generated by the Generator from noise z belongs to. $G(z)$ is then passed to the Discriminator D_i selected by the Discriminator Selector, and D_i distinguishes between the dataset X_i of Class i and $G(z)$.

In this way, the Discriminator is responsible for identifying specific classes, which improves the overall discriminatory ability of the Discriminators, and the Generator's image generation ability is further improved by adversarial training with these Discriminators.

4 CLASS ASSIGNMENT FOR MULTIPLE DISCRIMINATORS

When using multiple specialized Discriminators to determine the authenticity of images generated by a Generator, it is essential to assign each Discriminator a specific image class (or domain) to identify. In this research, we propose two distinct methods for assigning image classes to each Discriminator. The following sections will provide a detailed explanation of these two methods.

4.1 Class Assignment Based on Hand-Defined Classes

First, we describe a method for assigning classes to multiple Discriminators based on hand-defined classes. In image generation tasks, it is sometimes possible to roughly categorize the images we aim to generate based on human senses. In such cases, the method described in this section is useful.

Consider, for example, the task of generating face images. In this scenario, we can divide the face images into some classes based on human senses. For instance, we might set up four hand-defined classes: male children, female children, adult men, and adult women. Then, for assigning each class image to an individual Discriminator, we need to divide the image dataset into these multiple classes. In this research, we use CLIP (Radford et al., 2021), a model trained on a large dataset of text and images, to obtain the relationship between each class label and its corresponding set of images in a zero-shot manner.

Specifically, as shown in the upper part of Figure 3, we use CLIP Text Encoder to compute the text features T_i for the hand-defined N classes, Class i $(1, \cdots, N)$. Next, we select an image x_j $(j = 1, \cdots, M)$ from the image dataset X that comprises the images we aim to generate and calculate its image feature I_j using CLIP Image Encoder. Then, by selecting the text feature T_i that best matches I_j based on cosine similarity, we assign image x_j to Class i. By repeating this process for all images in dataset X, we partition X into N class-specific subsets X_i $(i = 1, \cdots, N)$, as shown in the lower part of Figure 3.

In our research, the N Discriminators are each responsible for determining the authenticity of images belonging to one of these N classes. This approach leverages hand-defined categories combined with automatic class assignment using CLIP, enabling each Discriminator to specialize in judging a specific subset of images, thus enhancing the Generator's performance across diverse classes.

4.2 Class Assignment Based on Data Distribution

We next describe a method for assigning classes to each Discriminator based on the distribution of image features within the training data. This method automatically determines the optimal class distinction and allows each discriminator to specialize on a subset of images with similar characteristics. By using data-driven clustering, each Discriminator can learn to distinguish its assigned classes more effectively, which improves the overall performance of the Generator.

In this method, the image dataset X is automatically divided into N datasets X_i $(i = 1, \cdots, N)$ based on its distribution. First, the M images x_i $(i = 1, \cdots, M)$ in the training dataset are input into CLIP Image Encoder, resulting in M image feature vectors I_i $(i = 1, \cdots, M)$. Then, as an initial state for clustering, the M image feature vectors I_i are treated as M clusters C_i $(i = 1, \cdots, M)$ in the image feature space,

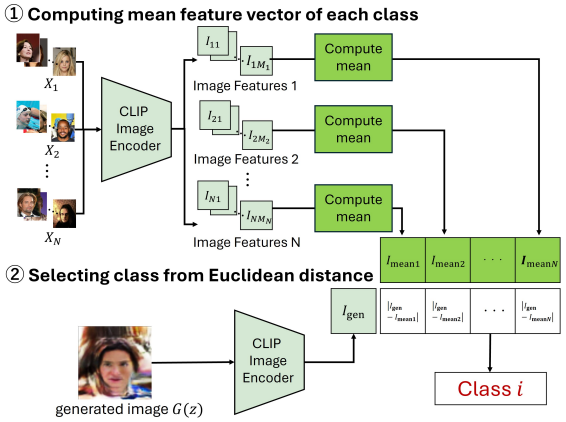


Figure 4: CLIP-based Discriminator Selector.

thereby forming the cluster set C as follows:

$$C = \{C_1, C_2, \ldots, C_M\} \tag{1}$$

The distance $d(C_i, C_j)$ between clusters C_i and C_j is defined based on Ward's method (Ward, 1963) as follows:

$$d(C_i, C_j) = \frac{|C_i||C_j|}{|C_i| + |C_j|} \|\mu_i - \mu_j\|^2 \tag{2}$$

where, $|C_i|$ represents the number of data points in cluster C_i, μ_i is the centroid (mean vector) of cluster C_i, and $\|\mu_i - \mu_j\|^2$ denotes the squared Euclidean distance between the centroids of clusters C_i and C_j respectively.

Then, the two clusters C_i and C_j with the smallest distance d are merged to form a new cluster C_{ij}.

$$C_{ij} = C_i \cup C_j \tag{3}$$

As a result, the new cluster set C is generated as follows:

$$C \leftarrow (C \setminus \{C_i, C_j\}) \cup \{C_{ij}\} \tag{4}$$

By repeating this operation until the number of clusters reaches N, the image dataset X can be divided into N clusters X_i $(i = 1, \cdots, N)$ based on the distribution of image features obtained from CLIP.

5 DISCRIMINATOR SELECTION

Next, we describe a method for selecting which of the N Discriminators should handle a generated image $G(z)$ from the Generator. In this research, we introduce a new component, the Discriminator Selector, responsible for making this selection. This Selector identifies the appropriate Discriminator (and thus the appropriate class) for each generated image based on its image features. We implement the Discriminator Selector using CLIP.

As shown in Figure 4, for each of the N classes defined in Section 4, we convert the images $x_{ij} \in X_i$ $(i = 1, \cdots, N)$ into image feature vectors I_{ij} using CLIP Image Encoder. We then compute a mean feature vector I_{mean_i} for each class as follows:

$$I_{\text{mean}_i} = \frac{1}{M_i} \sum_{j=1}^{M_i} I_{ij} \tag{5}$$

where, M_i denotes the number of data in Class i.

Then, as shown in Figure 4, the class c to which the generated image I_{gen} belongs is determined by selecting the class with the minimum Euclidean distance between the feature vector of the generated image I_{gen} and the mean feature vector of each class. This is formulated as follows:

$$c = \underset{i \in \{1,\ldots,N\}}{\arg\min} \|I_{\text{gen}} - I_{\text{mean}_i}\| \tag{6}$$

Each time the Generator produces an image, the Discriminator Selector uses this criterion to assign the generated image to a specific class. Consequently, the corresponding Discriminator, specialized in that class, is selected to evaluate the generated image.

In this method, an appropriate Discriminator is automatically selected based on the characteristics of each generated image, resulting in improved discriminative capability compared to using a single Discriminator. Consequently, this enhanced discrimination leads to improved generative capabilities of the Generator, as it receives more targeted and specialized feedback from the selected Discriminator.

6 GRADUAL INCREASE OF DISCRIMINATORS

In this section, we describe an approach for more efficient and stable training when using multiple Discriminators. When assigning distinct specializations to each Discriminator for adversarial learning, the initial stages of training can be challenging. At this early stage, the Generator's image quality is often low, making it difficult to assign generated images to the most suitable Discriminator (i.e., to determine the correct class). This poses a stability issue in the early training phases. Additionally, optimizing multiple Discriminators sequentially can be time-consuming.

To address these challenges, our approach gradually increases the number of Discriminators over the course of training. In the initial stage, we start with one Discriminator adversarially trained against a single Generator. Once the training has progressed to a

certain extent, we incrementally increase the number of Discriminators. Each time we increase the number of Discriminators, we do so by splitting each existing Discriminator into two. Thus, the number of Discriminators increases as 1, 2, 4, 8, and so forth. For each split, the final parameters of the original Discriminator are used as the initial parameters for the two new Discriminators. This inheritance of parameters ensures that the characteristics of the original Discriminator are retained in the split Discriminators, allowing for stable handling of the expansion process.

There are several possible methods for determining the timing of Discriminator increases. In this research, for simplicity, we chose to increase the number of Discriminators after a fixed number of training iterations. Developing a method to automatically determine this timing is an area for future investigation.

With this approach, we can avoid instability in Discriminator selection during the early stages of training. Additionally, as adversarial learning progresses more efficiently, this method enables performance improvements in the Generator with fewer training iterations.

7 TRAINING

Next, we discuss the training method for the network. Here, we consider the case of adversarially training a single Generator and N Discriminators D_i $(i = 1, \cdots, N)$ using a training dataset divided into N classes.

Suppose that an image $G(z)$ generated by the Generator from noise z is assigned to a Discriminator D_i by the Discriminator Selector. In this scenario, the Generator G and the selected Discriminator D_i engage in adversarial training according to the following objective functions:

$$G^* = \arg\min_G D_i \tag{7}$$

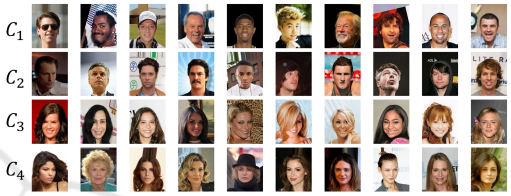
$$D_i^* = \arg\max_{D_i} \mathbb{E}_{x\sim p_{\text{data}}(x)}[\log D_i(x)] + \mathbb{E}_{z\sim p_z(z)}[\log(1 - D_i(G(z)))] \tag{8}$$

In this way, an appropriate Discriminator is selected by the Discriminator Selector based on the characteristics of the image generated by the Generator, enabling targeted adversarial learning to take place.

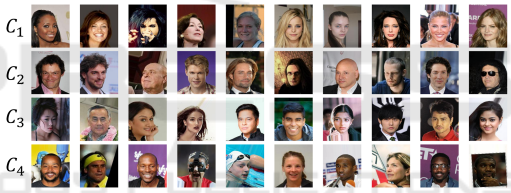
8 EXPERIMENTS

Next, we present the experimental results using the proposed method. In the following experiments, we target the task of generating face images using the CelebA (Liu et al., 2015) dataset shown in Figure 5 as training data. In each experiment, the dataset was divided using the proposed method, and a discriminator was selected using a Discriminator Selector to train the GAN. For quantitative evaluation, we used Fréchet Inception Distance (FID) (Heusel et al., 2017) to measure the similarity in the feature space between the generated images and the real images.

Figure 5: CelebA dataset(64×64 resized).



(a) Training images classified by hand-defined classes

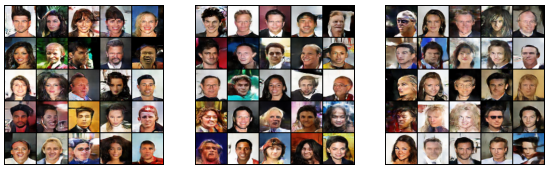


(b) Training images classified by image distribution

Figure 6: Comparison of training images classified into 4 classes C_i $(i = 1, \cdots, 4)$ by hand-defined classes and image distribution.

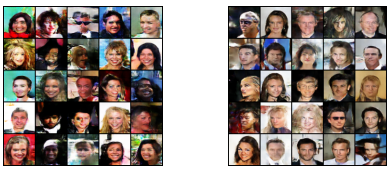
8.1 Image Generation Accuracy of Generator

In this experiment, we compared the image generation accuracy of the generator by setting the number of discriminators to 1, 2, and 4. Of these three cases, the case with 1 discriminator is the conventional method.



(a) 1G1D (b) 1G2D (c) 1G4D

Figure 7: Images generated from 1G1D, 1G2D, and 1G4D.



(a) hand-defined (b) data distribution

Figure 8: Images generated from 1G4D trained by using the hand-defined classes and the data distribution based classes.

For dataset division based on hand-defined classes, we set the following four classes and divided the dataset using CLIP Text Encoder and Image Encoder.

- ”Man with a smile”
- ”Man with a Neutral-face”
- ”Woman with a smile”
- ”Woman with a Neutral-face”

Example images in the four classes after division are shown in Figure 6 (a). Each row is a group of images corresponding to ”Man with a smile”, ”Man with a Neutral-face”, ”Woman with a smile”, and ”Woman with a Neutral-face” respectively. From these images, we find that by using CLIP Encoder, the images can be roughly classified correctly into the four classes.

Next, we show the results of automatic data segmentation based on the distribution of the image features in Figure 6 (b). Each of the four rows represents one of four classes of images after segmentation. From the top row, these images appear to be roughly divided into Western women, Western men, Asians, and Athletes. These results are different from the hand-defined classes in Figure 6 (a), but we find that the images have been segmented into groups with similar image features.

Based on these data classifications, we compared the image generation ability of the generator when adversarial learning was performed between one generator and four discriminators (1G4D) with adversarial learning with one discriminator (1G1D) and adversarial learning with two discriminators (1G2D). In the case of 1G2D, we transitioned from 1G1D to 1G2D in 50 epochs. In the case of 1G4D, we transitioned from 1G1D to 1G2D in 50 epochs, and from 1G2D to 1G4D in 100 epochs.

Table 1: FID score of generated images by the Generator trained with each configuration.

	hand-defined	data distribution
1G1D	95.71	94.74
1G2D	90.33	89.85
1G4D	90.03	87.37

Table 2: FID score of generated images by the Generator trained with each configuration.

	FID
1G4D (4)	failed
1G4D (1,4)	96.42
1G4D (1,2,4)	87.37

Figure 7 shows example images generated from 1G1D, 1G2D, and 1G4D respectively. Figure 8 also shows a comparison of images generated from 1G4D using hand-defined classes and using data distribution based classes.

To quantitatively evaluate the quality of these generated images, we computed the FID values of the generated images. The results are shown in Table 1.

Table 1 shows that 1G2D and 1G4D can generate generators with higher performance than the conventional method 1G1D, and that the more discriminators are added, the higher the performance of the generator. Also, comparing the second and third columns of this table, we find that automatic data classification based on the distribution of the data can generate a generator with higher performance than the hand-defined classes.

8.2 Gradual Increase of Discriminators

Next, we investigated the difference in image generation results between learning while gradually increasing the number of discriminators and learning with an increased number of discriminators from the beginning of learning. We compared three cases: (1) 1G4D from the beginning (1G4D (4)), (2) starting with 1G1D and changing to 1G4D at the 50th epoch (1G4D (1,4)), and (3) starting with 1G1D, changing to 1G2D at the 50th epoch and changing to 1G4D at the 100th epoch (1G4D (1,2,4)). The FID score of the generator trained in each configuration is shown in Table 2.

In the case of IG4D (4), the image generation by the Generator in the early stages of learning was low accuracy, so the discriminator selection could not be performed appropriately and training failed. On the other hand, comparing the results of 1G4D (1,4) and 1G4D (1,2,4), we find that 1G4D (1,2,4) has a higher generation accuracy of the Generator, and that gradually increasing the number of discriminators can gen-

erate a better Generator.

These results show that when using multiple discriminators, gradually increasing their expertise can result in better adversarial learning.

9 CONCLUSION

In this paper, we proposed a method to improve the image generation ability of a generator by adversarially training a generator using multiple discriminators with different expertise. In particular, we proposed a method to give multiple discriminators independent expertise by dividing a dataset so that they have independent image features and selecting images for each discriminator by using CLIP. In addition, we showed a method to gradually increase the number of discriminators in order to eliminate the instability of training that occurs when using multiple discriminators.

Experimental results showed that when dividing the classes that each discriminator is responsible for, more appropriate expertise is given to the discriminators when dividing based on the distribution of image features obtained by CLIP rather than based on features thought by humans. It was also revealed that it is more efficient to give expertise while gradually increasing the number of discriminators.

REFERENCES

Albuquerque, I., Monteiro, T., Doan, T., Considine, B., Falk, T., and Mitliagkas, I. (2019). Multi-objective training of generative adversarial networks with multiple discriminators. In *Proc. International Conference on Machine Learning*.

Brock, A., Donahue, J., and Simonyan, K. (2019). Large scale gan training for high fidelity natural image synthesis. In *International Conference on Learning Representations (ICLR)*.

Choi, J. and Han, B. (2022). Mcl-gan:generative adversarial networks with multiple specialized discriminators. In *Proc. Conference on Neural Information Processing Systems (NeurIPS)*.

Dinh Nguyen, T., Le, T., Vu, H., and Phung, D. (2017). Dual discriminator generative adversarial nets. In *Proc. Conference on Neural Information Processing Systems (NIPS)*.

Durugkar, I., Gemp, I., and Mahadevan, S. (2017). Generative multi-adversarial networks. In *Proc. International Conference on Learning Representations*.

Ghosh, A., Kulharia, V., Namboodiri, V., Torr, P., and Dokania, P. (2018). Multi-agent diverse generative adversarial networks. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014). Generative adversarial nets. *Advances in neural information processing systems*, 27:2672–2680.

Guzman-Rivera, A., Batra, D., and Kohli, P. (2012). Multiple choice learning: Learning to produce multiple structured outputs. In *Proc. Conference on Neural Information Processing Systems (NIPS)*.

Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., and Hochreiter, S. (2017). Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in Neural Information Processing Systems*, 30.

Hoang, Q., Nguyen, T., Le, T., and Phung, D. (2018). Mgan: Training generative adversarial nets with multiple generators. In *Proc. International Conference on Learning Representations*.

Karras, T., Laine, S., and Aila, T. (2020). Analyzing and improving the image quality of StyleGAN. In *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8110–8119.

Kingma, D. P. and Welling, M. (2014). Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*.

Ledig, C., Theis, L., Huszár, F., Caballero, J., Cunningham, A., Acosta, A., Aitken, A., Tejani, A., Totz, J., Wang, Z., and Shi, W. (2017). Photo-realistic single image super-resolution using a generative adversarial network. *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4681–4690.

Liu, Z., Luo, P., Wang, X., and Tang, X. (2015). Deep learning face attributes in the wild. In *Proc. International Conference on Computer Vision (ICCV)*.

Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G., and Sutskever, I. (2021). Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning (ICML)*. PMLR.

Ramesh, A., Pavlov, M., Goh, G., Gray, S., Voss, C., Radford, A., Chen, M., and Sutskever, I. (2021). Zero-shot text-to-image generation. *arXiv preprint arXiv:2102.12092*.

Rombach, R., Blattmann, A., Lorenz, D., Esser, P., and Ommer, B. (2022). High-resolution image synthesis with latent diffusion models. In *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10684–10695.

Ward, J. H. (1963). Hierarchical grouping to optimize an objective function. *Journal of the American Statistical Association*, 58(301):236–244.

Zhang, H., Xu, T., Li, H., Zhang, S., Huang, X., Wang, X., and Metaxas, D. N. (2017). Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks. In *Proc. IEEE International Conference on Computer Vision (ICCV)*, pages 5907–5915.