



STIMA / IDA Department

RESEARCH PROJECT

RankMe for Detecting Mode-Drop in Generative AI

Arash Haratian

2023-2024

# 1. Abstract

The field of Generative AI is experiencing a surge in popularity as they can produce text, images, and videos. However, the vast dimensionality of the domain presents challenges when it comes to evaluating the model [Gozalo-Brizuela and Garrido-Merchán, 2023]. Ideally, a model should be capable of generating data across the entire domain from which it was trained, but these models often have the problem of forgetting some of the data (mode-drop) or they generate data outside of the domain (hallucination). These problems may result in biased or false generated data [Gozalo-Brizuela and Garrido-Merchán, 2023].

In the recent research paper from Garrido et al. [2023], they introduce RankMe method, a promising solution for assessing representations. This paper inspired this research project where the aim is to investigate the usage of RankMe method on the problem of mode-drop and see if it can be a promising way to detect mode-drop or not. We notice that RankMe can detect the mode-drop in most of the cases where other existing methods may not be able to.

## 2. Introduction and Background

Generative models are one of the most promising approaches towards unsupervised learning techniques in machine learning. They are trained on a large amount of data in some domain (e.g., millions of images, sentences, or sounds, etc.) and then trained to generate data similar to the training data [Karpathy et al., 2016]. However, they often suffer from problems like mode-drop and mode-collapse.

Mode-drop is a common problem in Generative models and refers to the reduced variety in the samples produced by a generator. The term has been used interchangeably and has two main interpretations [Valle, 2019]:

- The generator synthesizes samples with intra-mode variety, but some modes are missing.
- The generator synthesizes samples with inter-mode variety, but each mode lacks variety.

Both interpretations assume that a model has high precision and low recall. The first interpretation refers to low inter-mode recall, that is, when there is a variety in the modes present in the synthesized samples but not all modes are present. The second interpretation refers to low intra-mode recall, that is, when there is a variety within each mode but the overall variety of modes is low [Valle, 2019].

There have been numerous works addressing these issues. For instance, various training techniques and modifications to the Generative models architecture have been proposed to mitigate mode-drop. Despite these efforts, mode-drop remains a challenging issue in the field of generative AI. [Varshney et al., 2020, Yazici et al., 2020, Dash et al., 2021]

The paper “RankMe: Assessing the Downstream Performance of Pretrained Self-Supervised Representations by Their Rank” by Garrido et al. [2023] focuses on Joint-Embedding Self Supervised Learning (JE-SSL). JE-SSL has seen rapid development but lacks principled guidelines for successful deployment [Chen et al., 2020, He et al., 2020, Li et al., 2022, Zhou et al., 2022, Lee et al., 2022, Bardes et al., 2022, Caron et al., 2021]. This is mainly because JE-SSL does not employ any input reconstruction, thus lacking visual cues of unsuccessful training.

To address this, the authors developed RankMe, a simple unsupervised criterion that indicates the quality of the learned JE-SSL representations by their effective rank. RankMe is computationally friendly and does not require any labels to assess the performance of JE-SSL representations, even on different downstream datasets. Moreover, RankMe does not have any training or hyperparameters to tune. The authors hope that RankMe will facilitate the deployment of JE-SSL towards domains that do not have the opportunity to rely on labels for representations’ quality assessment. The problem of mode-drop can be seen as such problems where the generated data has no label to indicate which of the classes and attributes of the training data points are presented in the newly generated data.

It is expected that the representations of the data that are generated from a Generative model get a high score from RankMe method. In other words, the rank of the representations of a biased generated data (due

to the mode-drop) should be lower than the rank of the representations of an unbiased generated data (no mode-drop).

In this study, the

### 3. Methods

In general the following is the outline of the research project:

1- We need some generated images. For this purpose, the CelebA dataset by Liu et al. [2015] is selected. The dataset consists of 202599 generated images of human faces where each image has 40 binary attributes (such as Male, Attractive, Young, etc.).

2- Having the image dataset, an encoder model is needed to get the representation of the generated images. Based on the results of the RankMe original paper, VicReg [Bardes et al., 2022] and Dino[caron2021emerging] are selected and both are tested for our study. Both representation learning models create a representation vector of 2048 for each input image.

3- Some evaluation metrics should be selected as baselines. In this study Inception score (IS) [Salimans et al., 2016] is selected. Higher Inception score shows that the Generative model is less prone to mode-drop. notice that high RankMe is better

4- A pipeline for evaluating each of the baseline metrics and RankMe is needed. The pipeline should be able to generate a biased dataset that has some of the attributes missing (have mode-drop), then generates the representations of the images using VicReg or Dino, and finally calculates the scores from each of the metrics. Having binary attributes for CelebA dataset will be useful for excluding or including attributes to biased dataset.

A Singularity container [Kurtzer et al., 2017] is also has been made to make the results of this project to be more reproducible.

In order to be able to use RankMe correctly, the formula of the RankMe should be studied first. Garrido et al. [2023] introduce RankMe formally as the following smooth rank measure where it is originally introduced in [Roy and Vetterli, 2007]:

$$RankMe(Z) = \exp \left( \sum_{k=1}^{\min(N,K)} p_k \log p_k \right)$$

$$\text{and } p_k = \frac{\sigma_k(Z)}{\|\sigma(Z)\|_1} + \epsilon$$

Where  $Z$  is a  $(N \times K)$  representation matrix of the input data, and the  $\sigma_k$  is the  $k^{th}$  singular value of the  $Z$  matrix. Notice that  $N$  is the number of observations and  $K$  is the size of representation vector the network. Based on the information from the formula, our problem needs the representation of the generated images. After that the RankMe score can be obtained.

Before discussing the results of the pipeline, the basic python implementation of the RankMe in PyTorch can be seen as follows:

```
## GETTING THE REPRESENTATION VECTORS
with torch.no_grad():
    for batch, x in enumerate(data_loader):
        model_output[(batch*batch_size):((batch+1)*batch_size), :] = model(x)

## CALCULATING THE SINGULAR VALUES
sigma = calc_singular(model_output)
```

```

eps = torch.finfo(model_output.dtype).eps

## CALCULATING THE  $p_k$  IN THE ABOVE FORMULA
p_ks = (sigma / torch.abs(sigma).sum() ) + eps

## CALCULATING THE RankMe SCORE
rank = torch.exp(-torch.sum(p_ks * torch.log(p_ks)))
rank = rank.unsqueeze(0)

```

In order to make sure the RankMe score is implemented correctly, the figure S11 from the original paper is re-implemented in the supplement and the result can be seen in Figure 1.

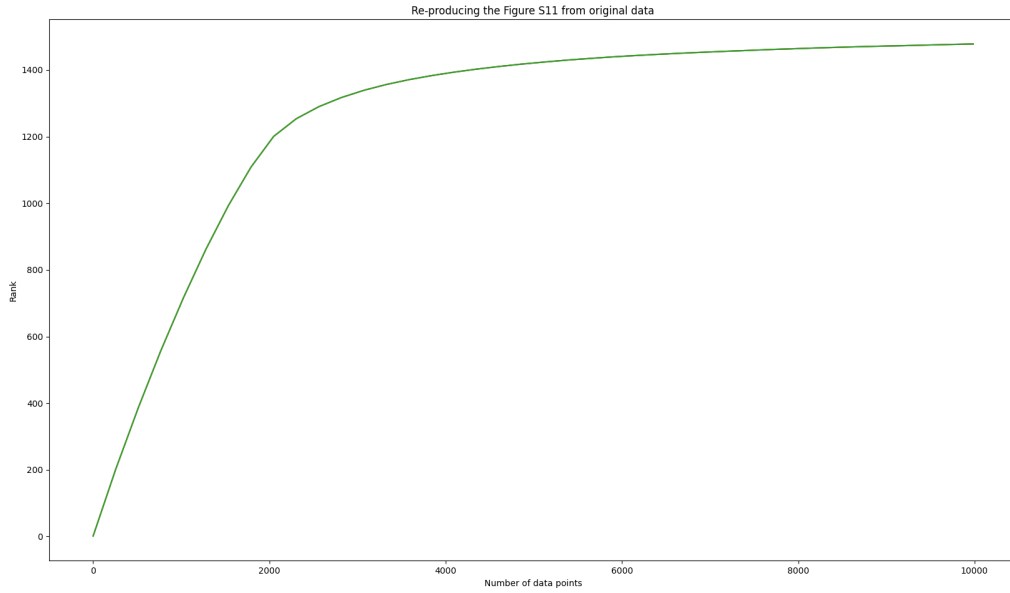


Figure 1: The re-implementation of the Figure S11 from the original RankMe paper. Notice that in the original paper, the value on y-axis is a ratio while this plot is the rank of the matrix itself. This plot shows that the rank of the matrix will be calculated correctly as the amount of data increases. In other word in converges to the true rank of the matrix.

## 4. Results and Discussions

The pipeline is ran with two different settings, once it just creates biased datasets by just dropping “Male” attribute, and once dropping multiple attributes (“Attractive”, “High Cheekbones”, “Young”). Then, the biased datasets are passed to both VicReg and Dino to get their representations. The RankMe and IS are computed on the obtained representations. Notice that the same pipeline also can be used to calculate the scores for unbiased datasets (where the target attributes are presented equally). Each of the described settings is ran for five times with different random seeds. The seeds that are used for these results are 1, 2, 3, 4, 5.

First, the results of the RankMe and IS on the setting where only one attribute (“Male”) is dropped, can be seen in Figure 2 and Figure 3. In this case, biased datasets and unbiased datasets contain 51,200 images. As it is evidence in these plots, RankMe score for the unbiased datasets is higher than the datasets that have one

attribute dropped. This is while the IS score cannot detect the mode-drop as its scores for all the datasets are very close to each other.

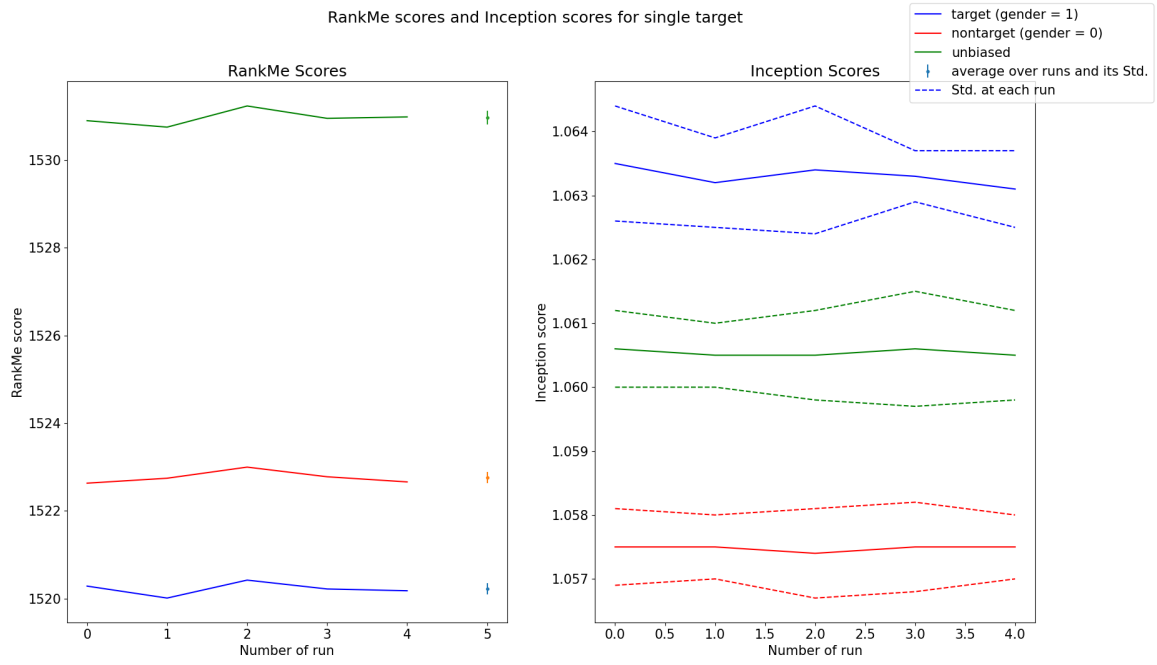


Figure 2: The results of the VicReg + RankMe score and VicReg + IS score when one of the attributes is dropped.

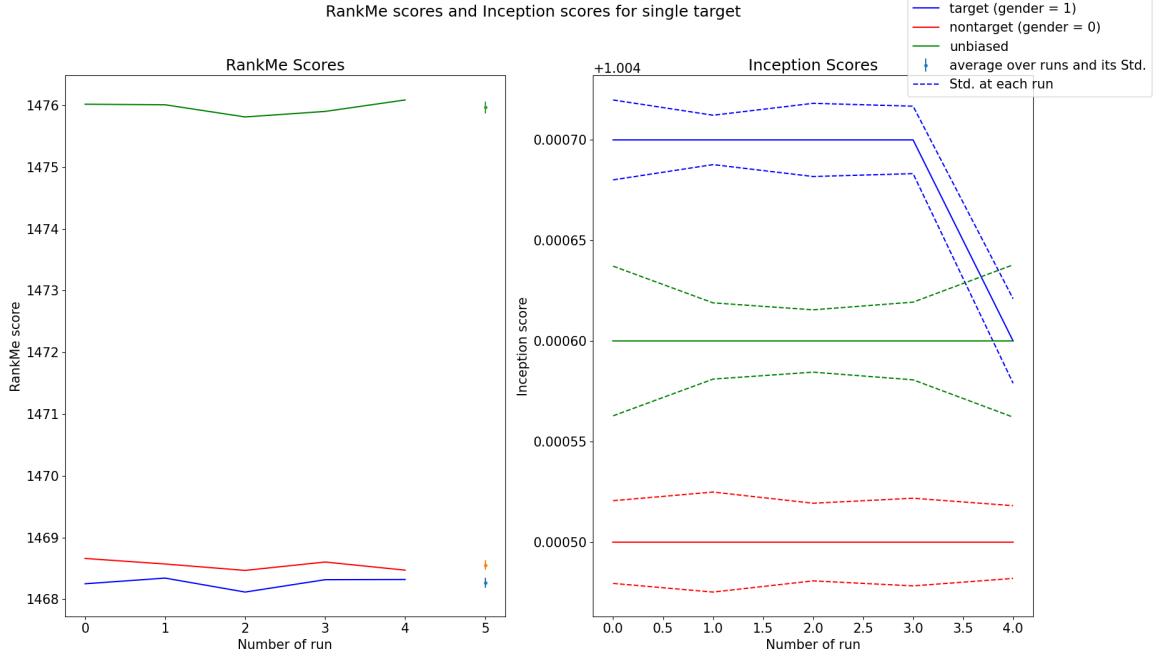


Figure 3: The results of the Dino + RankMe score and Dino + IS score when one of the attributes is dropped.

Now, the results of the RankMe and IS on the setting where three attributes (“Attractive”, “High Cheekbones”, “Young”) are dropped, can be seen in Figure 4 and Figure 5. In this case, biased datasets and unbiased datasets contain 21,760 images. In the same way for the Figure 2 and 3, RankMe score can detect the mode-drop while IS score still cannot detect the mode-drop. Notice that in Figure 5, the empirical standard deviation of Dino + RankMe is calculated over 5 runs, and it is visible that the one standard deviations of the scores for “nontarget dataset” and “unbiased dataset” are overlapping.

Overall, the RankMe can be a new metric for the field of Generative models for detecting the mode-drop as long as the representation network that is chosen is good. In our study case, VicReg and RankMe are more confident at detecting mode-drop compared to the Dino and RankMe.

## 4.2 Suggested Research

One bottleneck in this study is long running time of the pipeline. The RankMe paper mentions that in order to get the best estimation of the rank of the representations, a huge amount of data is needed. In fact, it can be seen in Figure 1 or S11 of the original RankMe paper that as the amount of data increases the RankMe formula gets closer to the rank of the representation matrix. One suggestion for further studying is to investigate if it is possible to calculate singular values on batches of representations and then using them inside the summation in the above formula or not. Currently, the implemented code, gathers all the representations of the generated dataset and then it calculates the singular values which may not be the most efficient way. However, it should be mentioned that the images of CelebA dataset are large, and hence they could contribute to the long running time. One suggestion is that, CIFAR10 or CIFAR100 by Krizhevsky [2009] could be used in the same way as CelebA is used in this study but with the difference that the CIFAR10 and CIFAR 100 have real data images.

Moreover, there can be more studies on whether RankMe can detect hallucination or mode-collapse or not which are outside of the scopes of this study.

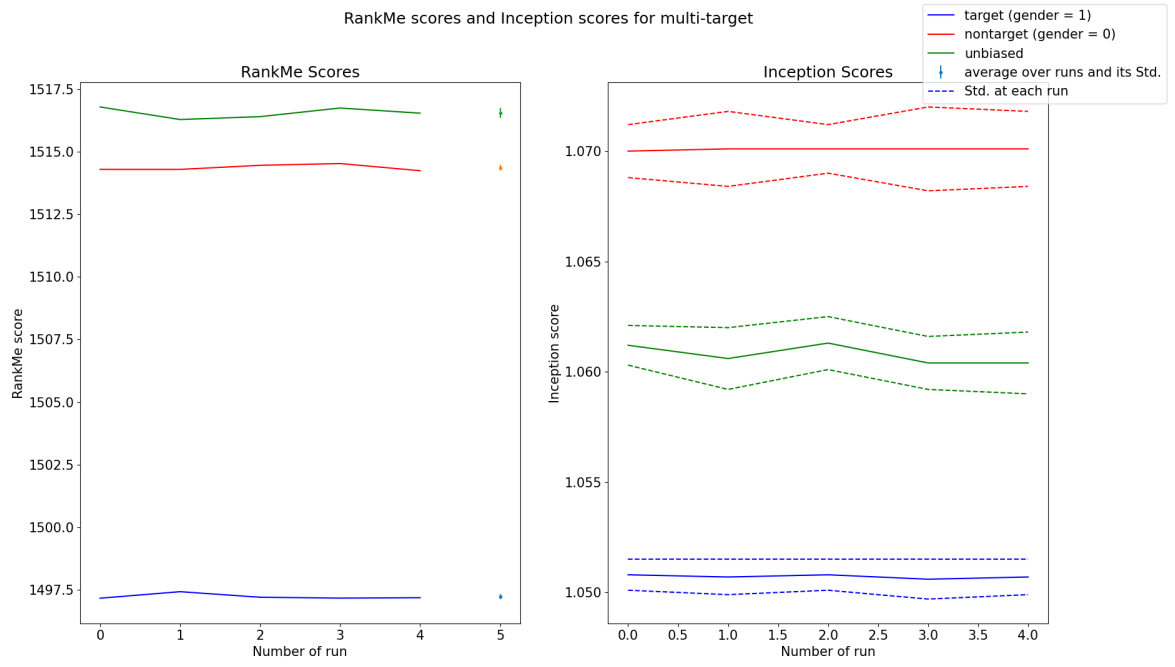


Figure 4: The results of the VicReg + RankMe score and VicReg + IS score when three of the attributes are dropped.

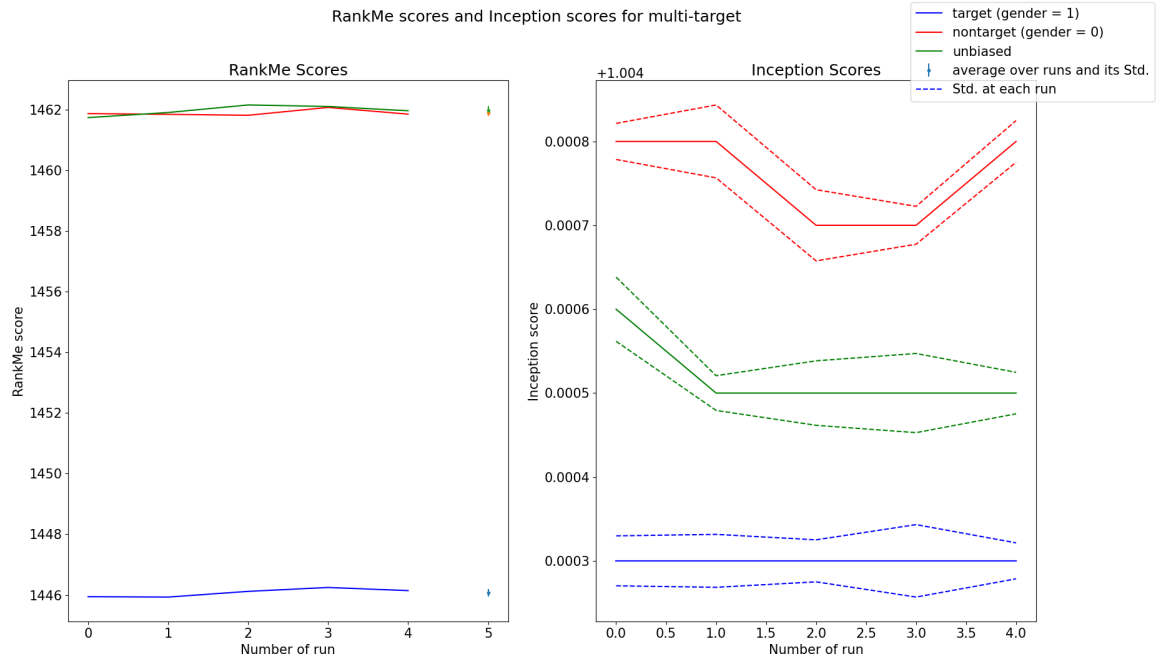


Figure 5: The results of the Dino + RankMe score and Dino + IS score when three of the attributes are dropped.

## 5. References

- Adrien Bardes, Jean Ponce, and Yann LeCun. Vicreg: Variance-invariance-covariance regularization for self-supervised learning, 2022.
- Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers, 2021.
- Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations, 2020.
- Ankan Dash, Junyi Ye, and Guiling Wang. A review of generative adversarial networks (gans) and its applications in a wide variety of disciplines – from medical to remote sensing, 2021.
- Quentin Garrido, Randall Balestriero, Laurent Najman, and Yann Lecun. Rankme: Assessing the downstream performance of pretrained self-supervised representations by their rank, 2023.
- Roberto Gozalo-Brizuela and Eduardo C. Garrido-Merchán. A survey of generative ai applications, 2023.
- Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning, 2020.
- Andrej Karpathy, Pieter Abbeel, Greg Brockman, Peter Chen, Vicki Cheung, Yan Duan, Ian Goodfellow, Durk Kingma, Jonathan Ho, Rein Houthooft, Tim Salimans, John Schulman, Ilya Sutskever, and Wojciech Zaremba. Generative models, 2016. URL <https://openai.com/research/generative-models>.
- Alex Krizhevsky. Learning multiple layers of features from tiny images. pages 32–33, 2009. URL <https://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf>.
- Gregory M. Kurtzer, Vanessa Sochat, and Michael W. Bauer. Singularity: Scientific containers for mobility of compute. *PLOS ONE*, 12(5):1–20, 05 2017. doi: 10.1371/journal.pone.0177459. URL <https://doi.org/10.1371/journal.pone.0177459>.
- Youngwan Lee, Jeffrey Willette, Jonghee Kim, Juho Lee, and Sung Ju Hwang. Exploring the role of mean teachers in self-supervised masked auto-encoders, 2022.
- Chunyu Li, Jianwei Yang, Pengchuan Zhang, Mei Gao, Bin Xiao, Xiyang Dai, Lu Yuan, and Jianfeng Gao. Efficient self-supervised vision transformers for representation learning, 2022.
- Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, December 2015.
- Olivier Roy and Martin Vetterli. The effective rank: A measure of effective dimensionality. In *2007 15th European Signal Processing Conference*, pages 606–610, 2007.
- Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans, 2016.
- Rafael Valle. Hands-on generative adversarial networks with keras, 2019. URL <https://www.oreilly.com/library/view/hands-on-generative-adversarial/9781789538205/1133a516-b38d-4271-b510-0ba7077f7292.xhtml>.
- Sakshi Varshney, P. K. Srijith, and Vineeth N. Balasubramanian. Stm-gan: Sequentially trained multiple generators for mitigating mode collapse. In Haiqin Yang, Kitsuchart Pasupa, Andrew Chi-Sing Leung, James T. Kwok, Jonathan H. Chan, and Irwin King, editors, *Neural Information Processing*, pages 676–684, Cham, 2020. Springer International Publishing. ISBN 978-3-030-63823-8.
- Yasin Yazici, Chuan-Sheng Foo, Stefan Winkler, Kim-Hui Yap, and Vijay Chandrasekhar. Empirical analysis of overfitting and mode drop in gan training, 2020.
- Pan Zhou, Yichen Zhou, Chenyang Si, Weihao Yu, Teck Khim Ng, and Shuicheng Yan. Mugs: A multi-granular self-supervised learning framework, 2022.