Generative Adversarial Networks for Clustering Semiconductor Wafer Maps

Hamidreza Mahyar TU Wien
Peter Tulala TU Wien
Hamid R. Rabiee Sharif University
Radu Grosu TU Wien

{hamidreza.mahyar, ptulala, radu.grosu}@tuwien.ac.at, rabiee@sharif.edu

Abstract

Semiconductor manufacturing processes are characterized by a certain amount of process deviations. Automated detection of these production issues followed by an automated root cause analysis has a potential to increase the effectiveness of semiconductor production. Manufacturing defects exhibit typical patterns in measured wafer test data. Recognizing these patterns is an essential step for finding the root cause of production issues. This paper demonstrates that combining Information Maximizing Generative Adversarial Network (InfoGAN) and Wasserstein GAN (WGAN) is suitable for extracting the most characteristic features from large real-world sensory wafer test data and in various aspects outperforms traditional unsupervised dimensionality reduction techniques. These features are then used in subsequent clustering task to group wafers into clusters according to the patterns they exhibit. The main outcome of this work is a statistical model for recognizing spatial patterns given a wafer map. We experimentally evaluate the performance of the proposed approach over a real dataset.

1 Introduction

Sustainable competitiveness in semiconductor industry requires a rapid development of increasingly complex semiconductor products, which drastically decreases the amount of time available for diagnosing production defects [1]. Semiconductor manufacturing is prone to production issues of two types – random defects or systematic defects. Random defects are usually attributed to the dust particles in the production environment and tend to be related to the overall cleanliness of the production environment. On the other hand, systematic defects are caused by a malfunction of a process equipment or human errors [2]. Due to hundreds of processing steps involved in semiconductor manufacturing, diagnosing wafers after each of these steps is not practical. Instead, equipment sensor values and electrical test data are collected only after most of the processing steps. It is assumed that systematic defects exhibit typical shapes in measured wafer test data (*e.g.* rings, spots, repetitive patterns, or scratches). Recognizing these patterns is an essential step for backtracking to which processing step caused the defects. Automated root cause analysis and decision-making with reduced human intervention has potential to significantly improve manufacturing efficiency of semiconductor industry. Therefore, proposing an efficient method for detecting systematic defects from given sensory data is a valuable contribution and inevitable task in accordance with this goal.

2 Related work

To address the aforementioned problem, several methods based on traditional image processing approaches have been proposed [3]. More robust methods utilized some machine learning techniques to recognize more complex patterns in wafer test data. There exist many methods based on supervised

32nd Conference on Neural Information Processing Systems (NIPS 2018), Montréal, Canada.

training of mixture models [4], singular value decomposition [5], neural networks [6] and support-vector machines [7]. Although these methods are powerful, their supervised nature still requires a human expert to craft a training dataset with manually labelled data. The apparent advantage of unsupervised approaches lies in the elimination of subjective factors from pattern recognition task, which in turn reduces costs and number of clustering errors. In the industry, it is required to automatically detect the hidden dependencies between different types of wafer defects without intervention of human expert which enables detection of patterns that were unknown or overlooked before. To this end, some methods have been proposed, such as self-organizing neural networks [8], self-organizing maps [9] as well as techniques based on dimensionality reduction like diffusion maps [10], discriminant analysis [11], and variational auto-encoder [12]. In this paper, we propose an unsupervised method for clustering wafer map patterns based on deep generative adversarial neural networks (GANs).

3 Proposed Approach

3.1 Data Pre-processing

Our available real dataset, provided by Infineon Technologies (`http://infineon.com`), consists of 6 wafer lots, each has 50 wafers containing 17509 chips. Each chip is measured with 20 different tests (features) and its position within a wafer is stored as a tuple. We consider each test measurement of a wafer as a bitmap. Overall, we have 6000 wafermaps, where each one represented as a bitmap of size 193x115 pixels. Data pre-processing is a fundamental step to clean the data before designing a machine learning model. We apply several consecutive pre-processing steps to raw wafermaps, which are depicted in Figure 1 [12]: (1) We utilize a median absolute deviation (MAD)-based outlier detection method by modifying the common Z-score mechanism [13]; (2) Wafermaps are binarized by replacing the present values with 1 and the missing values (holes) with 0. Mathematical morphology mechanism [14] is then used to close small holes in the wafer area and find contours of the wafer; (3) Missing values in wafer area are inpainted with values reconstructed from neighborhood information around each missing region, using Chui-Mhaskar inpainting algorithm [15] via solving the biharmonic equations; (4) After feature normalization, wafers are smoothened using the median filtering procedure within a sliding window. A sample wafer map and its pre-processed wafer map is depicted in Figure 2. The cleansed wafer maps can then be used for further tasks.

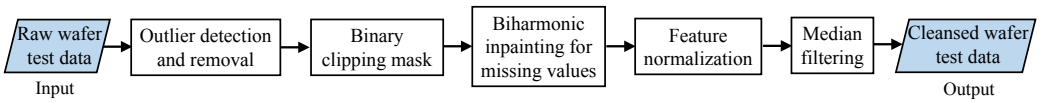


Figure 1: Wafer map pre-processing procedure

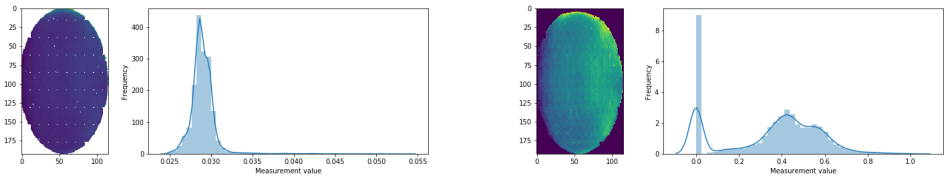


Figure 2: A raw wafer map (left) and its cleansed wafer map with clearly visible pattern (right).

3.2 A Generative Adversarial Network

The pre-processed wafer maps can be seen as N individual dataset containing iid samples from a real data distribution p_{data}. Now, our goal is feature extraction to overcome the *curse of dimensionality* [16] for the clustering task by extracting low-dimensional latent codes from high-dimensional cleansed wafer maps. Our approach is to use a deep generative adversarial neural network (GAN) [17, 18], which consists of two components: (1) The discriminator $D(.)$ estimates the probability of a given data sample x drawn from the real dataset with distribution p_{data}; (2) The generator $G(.)$ takes a latent code z sampled from a noise distribution p_{noise} and generates synthetic sample $G(z)$ as realistic as possible in order to fool the discriminator. The generator learns the distribution p_G that is an approximation of the real data distribution p_{data}. These two components are simultaneously trained to compete against each other. Samples from the real dataset and from

output of the generator are randomly passed to the discriminator. The objective of GAN can be then modelled as a two-player non-cooperative minimax game where each player attempts to optimize its own payoff with value function $V(D, G)$, as:

$$\min_G \max_D V(D, G) = \mathbb{E}_{x \sim p_{data}}[\log D(x)] + \mathbb{E}_{z \sim p_{noise}}[\log(1 - D(G(z)))] \tag{1}$$

This minimax game is useful for theoretical analysis of the problem, however it does not perform well in practice. When the learning process begins, D can simply rejects all generated samples with very high confidence and hence not provide sufficient gradient to improve the performance of G. As suggested in [17], instead of training G to minimize $\mathbb{E}_{z \sim p_{noise}}[\log(1 - D(G(z)))]$, we can train G to maximize $\mathbb{E}_{z \sim p_{noise}}[\log D(G(z))]$. Hence, two different loss functions for the discriminator and the generator can be used:

$$\begin{aligned} \mathcal{L}_D^{\text{GAN}} &= \mathbb{E}_{x \sim p_{data}}[\log D(x)] + \mathbb{E}_{z \sim p_{noise}}[\log(1 - D(G(z)))] \\ \mathcal{L}_G^{\text{GAN}} &= \mathbb{E}_{z \sim p_{noise}}[\log D(G(z))] \end{aligned} \tag{2}$$

An information-theoretic extension to GAN, known as InfoGAN [19], was proposed in order to learn disentangled representations in an unsupervised manner. InfoGAN extends the inputs of the generator by an additional *structured* latent code $c \sim p_{latent}$ (it can be extended to multiple structured latent codes). The minimax game for InfoGAN is formulated by adding a regularization term, as:

$$\min_G \max_D V'(D, G, Q) = V(D, G) - \lambda I(c; G(z, c)) \tag{3}$$

where $G(z, c)$ is a generator extended with a structured latent code c, $\lambda \in \mathbb{R}$ is a regularization coefficient and $I(\cdot)$ is mutual information between the structured latent code c and a sample drawn from $G(z, c)$. However, direct calculation of $I(c; G(z, c))$ requires an access to the intractable posterior $p(c|x)$. Instead, as shown in [19], we can obtain a lower bound of the mutual information by introducing an auxiliary distribution $q(c|x)$ to approximate $p(c|x)$, as follows:

$$\begin{aligned} I(c; G(z, c)) = \overbrace{H(c)}^{entropy} - \overbrace{H(c|G(z, c))}^{conditional\ entropy} &= \mathbb{E}_{x \sim G(z,c)}[H(c|x)] + H(c) \\ &\geq \mathbb{E}_{x \sim G(z,c)}[\mathbb{E}_{c' \sim p(c|x)}[\log q(c'|x)]] + H(c) \end{aligned} \tag{4}$$

Since c is sampled from a fixed latent code distribution, $H(c)$ can be treated as a constant. In practice, the auxiliary distribution $q(c|x)$ is parametrized by a neural network Q that shares all convolutional layers with D extended by one additional layer, hence it adds only a negligible computational cost. We used the normal distribution for the latent code distribution $q(c'|x)$ in our implementation, so:

$$I(c; G(z, c)) \geq \log\left(\frac{1}{\sqrt{2\pi\sigma^2}} \cdot e^{-\frac{(x-\mu)^2}{2\sigma^2}}\right) = -\frac{1}{2}\log(2\pi\sigma^2) - \frac{(x-\mu)^2}{2\sigma^2} = \mathcal{L}^{\text{InfoGAN}} \tag{5}$$

GANs are notoriously difficult to train. From a game-theoretic perspective, the generator and discriminator are trained to find a Nash equilibrium, however a convergence is not guaranteed due to non-cooperative nature of the minimax game [18]. There is an evidence that distributions p_G and p_{data} are concentrated on a low dimensional manifold with disjoint supports [20]. Vanishing gradients of the generator is another common problem as the performance of discriminator saturates [21]. Furthermore, the generator can learn to trick the discriminator by learning only a very small subset of the real dataset and producing samples with low variety. Although to handle the aforementioned challenges we used some techniques proposed in [18], we were not able to stabilize the training on our wafer dataset with loss function based on KL-divergence as used in the original GAN paper. To this end, the idea proposed in Wasserstein GAN (WGAN) [22] is based on replacing the loss function used in the original GAN by a distance measure called Earth Mover's (EM) distance or Wasserstein-1 distance. The EM distance between the real data and generator distributions is defined by:

$$W(p_{data}, p_G) = \inf_{\gamma \sim \Pi(p_{data}, p_G)} \mathbb{E}_{(x,y) \sim \gamma}[\|x - y\|] \tag{6}$$

where $\gamma(x, y)$ is a transport plan over all possible joint probability distributions $\Pi(p_{data}, p_G)$ between p_{data} and p_G. In other words, it indicates how much "*mass*" must be moved from x to y in order to transform the distribution p_{data} into the distribution p_G. Calculating all possible distributions is intractable, instead [22] proposed to use a minimax game based on Kantorovich-Rubinstein duality, as follows:

$$\min_G \max_D \mathbb{E}_{x \sim p_{data}}[f(D(x))] - \mathbb{E}_{\tilde{x} \sim p_G}[f(D(\tilde{x}))] \tag{7}$$

where the function $f : \mathbb{R} \to \mathbb{R}$ is a 1-Lipschitz continuous function satisfying $|f(x_1) - f(x_2)| \leq \gamma \cdot |x_1 - x_2|$ for every $x_1, x_2 \in \mathbb{R}$ and some real constant $\gamma \geq 0$. Arjovsky proposed to enforce 1-Lipschitz continuity, by clipping the discriminator parametrized by weights $w \in \mathcal{W}$, via clipping these weights to a small interval $w \in [-c, c]$ (for instance $c = 0.01$) after each gradient update:

$$\min_G \max_{w \in \mathcal{W}} \mathbb{E}_{x \sim p_{data}}[D_w(x)] - \mathbb{E}_{\tilde{x} \sim p_G}[D_w(\tilde{x})] \tag{8}$$

However, it is mentioned that weight clipping is a terrible way to enforce the 1-Lipschitz constraint. An improved way of enforcing 1-Lipschitz continuity was described in [23] by adding a gradient penalty regularization term to the original WGAN loss function, as:

$$\begin{aligned} \mathcal{L}_D^{\text{WGAN-GP}} &= \mathbb{E}_{x \sim p_{data}}[D(x)] - \mathbb{E}_{\tilde{x} \sim p_G}[D(\tilde{x})] + \alpha \mathbb{E}_{\hat{x} \sim p_{\hat{x}}}[(\|\nabla_{\hat{x}} D(\hat{x})\|_2 - 1)^2] \\ \mathcal{L}_G^{\text{WGAN-GP}} &= \mathbb{E}_{\hat{x} \sim p_G}[D(\hat{x})] = \mathbb{E}_{z \sim p_{noise}}[D(G(z))] \end{aligned} \tag{9}$$

where α is the regularization coefficient and $p_{\hat{x}}$ is a distribution laying between p_{data} and p_G, *i.e.* it can be sampled as $\hat{x} = \epsilon x + (1 - \epsilon)\tilde{x}$ for $\epsilon \sim U[0, 1]$.

Combining the WGAN and InfoGAN objectives, we propose the following loss function for the discriminator and generator to extract low-dimensional latent codes from high-dimensional wafermaps:

$$\begin{aligned} \mathcal{L}_D &= \mathcal{L}_D^{\text{WGAN-GP}} - \lambda \mathcal{L}^{\text{InfoGAN}} \\ \mathcal{L}_G &= \mathcal{L}_G^{\text{WGAN-GP}} \end{aligned} \tag{10}$$

$\mathcal{L}_D$ is a weighted sum multi-objective optimization function. As shown in [24], any convex Pareto optimal front can be obtained when α and λ are strictly positive.

The exact neural network architecture used in our implementation is depicted in Figure 3. Generator (depicted in blue) generates a wafer given random vectors z and c. Discriminator (depicted in blue and red) then decides whether the provided wafer image is true (given by the original wafer test data) or fake (generated). Classification network Q (depicted in orange) optimizes the mutual information between it's output distribution and the distribution of c that was provided to the generator. `LeakyReLU` activation function used after each fully connected inner layer. We used hyperbolic tangent (`tanh`) activation function for the generator output (since the wafer data are normalized to interval $[-1, 1]$). The first-order gradient-based optimizer `Adam` is used for the training phase.

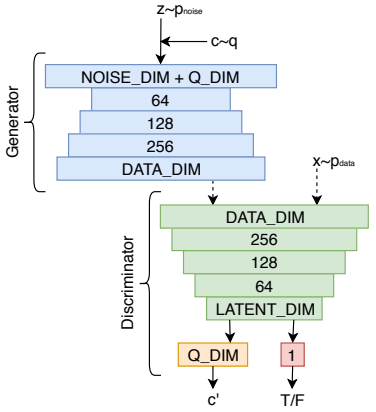


Figure 3: The proposed generative adversarial neural network.

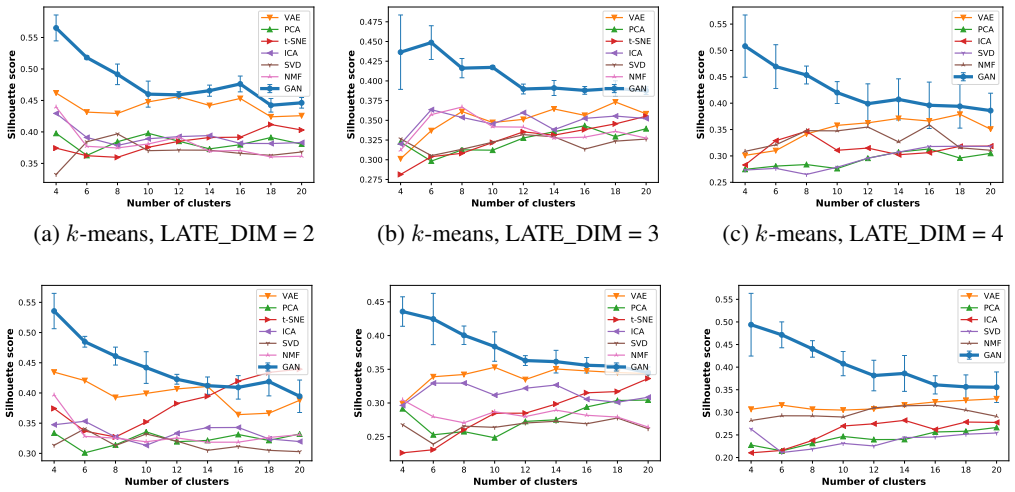
3.3 Wafermaps Patterns Clustering

We have described a mechanism for non-linear mapping of high-dimensional wafer measurement data into a low-dimensional representation. Now, we specify how to group the extracted latent features into clusters based on a distance measure. Wafermaps with similar patterns should be considered in the same cluster and dissimilar wafermaps should be clustered in different groups. There exist two types of clustering methods (*i.e.* hierarchical and partitioning) that can be applied for clustering of the wafermaps. We used one algorithm from each category, namely *Hierarchical agglomerative* and *k-means clustering*.

4 Experimental Evaluation

In this section, we evaluate the performance of the proposed GAN-based method compared to the other commonly used dimensionality reduction methods for spatial wafermaps patterns clustering. We developed all codes related to this work in Python v3.5.2 and the deep learning library Keras v2.0.8 with Tensorflow v1.5.0 backend. For comparison, we chose six well-known unsupervised feature extraction methods: (1)Variational auto-encoder (VAE) [12], (2) Non-negative matrix factorization (NMF) [25], (3) Singular value decomposition (SVD) [5], (4) Principal component analysis (PCA)

[26], (5) Independent component analysis (ICA) [27], (6) t-Distributed stochastic neighbor embedding (t-SNE) [28]. The clustering performance is measured with the Silhouette metric [29], that is a number on interval $[-1, 1]$ and defined as $sil(i) = \frac{b(i)-a(i)}{\max\{a(i),b(i)\}}$, where $a(i)$ is the average distance between feature vector x_i to the other vectors in the same clusters and $b(i)$ is the average distance between x_i to others in the nearest cluster. It shows how similar a latent feature is to other features within the same cluster compared to the other clusters. The higher this value is, the better the clustering performance will be. The average Silhouette values measured over all latent features with different dimensions (*i.e.* 2, 3, and 4) for our method and the competing methods are shown in Figure 4. For this experiment, we trained the model on 1000 epochs with batches of size 32. The results show that our approach outperforms the best existing methods for efficient clustering of spatial wafer map patterns in terms of Silhouette score, even in small number of clusters. One can easily see that our GAN-based method can get higher score in both partitioning and hierarchical clusterings in comparison with the competing methods. Moreover, variational auto-encoder has better performance in most of the cases among the competing methods.



(a) k-means, LATE_DIM = 2 (b) k-means, LATE_DIM = 3 (c) k-means, LATE_DIM = 4

(d) Agglom., LATE_DIM = 2 (e) Agglom., LATE_DIM = 3 (f) Agglom., LATE_DIM = 4

Figure 4: Evaluation of different feature extraction techniques in terms of Silhouette score with two clustering methods k-means and agglomerative clustering and different latent space dimensions. Our GAN-based approach yields better separated clusters compared to the competing methods in majority of cases.

5 Conclusion

Systematic defects in manufacturing industry are caused by a malfunction in a process equipment or human errors. Automated detection of such production issues and automated root cause analysis will improve the efficiency of semiconductor production. Manufacturing defects often exhibit patterns in measured test data. Recognizing these patterns and their categorization are essential tasks in root cause identification of the production issues. In this paper, we proposed a deep generative adversarial network methodology to recognize the spatial wafer map patterns. We extracted the most characteristic features of a large real sensory wafer test data, using a combination of the InfoGAN and the Wasserstein GAN. We then utilized the extracted features in clustering task to group the wafers into meaningful clusters based on their spatial patterns. Finally, we experimentally showed the superiority of the proposed approach over a real dataset, compared to the well-known methods.

Acknowledgement

The work has been performed in the project Power Semiconductor and Electronics Manufacturing 4.0 (SemI40), under grant agreement No 692466. The project is co-funded by grants from Austria, Germany, Italy, France, Portugal and Electronic Component Systems for European Leadership Joint Undertaking (ECSEL JU).

References

[1] Chia-Yu Hsu. Clustering ensemble for identifying defective wafer bin map in semiconductor manufacturing. *Mathematical Problems in Engineering*, 2015.

[2] Chih-Hsuan Wang. Recognition of semiconductor defect patterns using spectral clustering. In *Industrial Engineering and Engineering Management*, pages 587–591, 2007.

[3] Louis Breaux and Baljit Singh. Automatic defect classification system for patterned semiconductor wafers. In *Int Symp on Semiconductor Manufacturing*, pages 68–73, 1995.

[4] Jinho Kim, Youngmin Lee, and Heeyoung Kim. Detection and clustering of mixed-type defect patterns in wafer bin maps. *IISE Transactions*, 50(2):99–111, 2018.

[5] Kamal Taha, Khaled Salah, and Paul D Yoo. Clustering the dominant defective patterns in semiconductor wafer maps. *IEEE Transactions on Semiconductor Manufacturing*, 31(1):156–165, 2018.

[6] FL Chen, Sheng-Che Lin, K Yih-Yuh Doong, and KL Young. Logic product yield analysis by wafer bin map pattern recognition supervised neural network. In *Int Symp on Semiconductor Manufacturing*, 2003.

[7] Li-Chang Chao and Lee-Ing Tong. Wafer defect pattern recognition by multi-class support vector machines by using a novel defect cluster index. *Expert Systems with Apps*, 36(6):10158–10167, 2009.

[8] Chuan-Yu Chang, ChunHsi Li, Jia-Wei Chang, and MuDer Jeng. An unsupervised neural network approach for automatic semiconductor wafer defect inspection. *Expert Systems with Apps*, 36(1):950–958, 2009.

[9] Federico Di Palma, Giuseppe De Nicolao, Guido Miraglia, Egidio Pasquinetti, and Francesco Piccinini. Unsupervised spatial pattern classification of electrical-wafer-sorting maps in semiconductor manufacturing. *Pattern recognition letters*, 26(12):1857–1865, 2005.

[10] Gal Mishne and Israel Cohen. Multi-channel wafer defect detection using diffusion maps. In *IEEE Convention of Electrical & Electronics Engineers*, pages 1–5, 2014.

[11] Jianbo Yu and Xiaolei Lu. Wafer map defect detection and recognition using joint local and nonlocal linear discriminant analysis. *IEEE Transactions on Semiconductor Manufacturing*, 29(1):33–43, 2016.

[12] Peter Tulala, Hamidreza Mahyar, Elahe Ghalebi, and Radu Grosu. Unsupervised wafermap patterns clustering via variational autoencoders. In *IJCNN, Rio de Janeiro, Brazil*. 2018.

[13] Christophe Leys, Christophe Ley, Olivier Klein, Philippe Bernard, and Laurent Licata. Detecting outliers: Do not use standard deviation around the mean, use absolute deviation around the median. *Journal of Experimental Social Psychology*, 49(4):764–766, 2013.

[14] Chris Solomon and Toby Breckon. *Fundamentals of Digital Image Processing: A practical approach with examples in Matlab*. John Wiley & Sons, 2011.

[15] Charles K Chui and HN Mhaskar. Mra contextual-recovery extension of smooth functions on manifolds. *Applied and Computational Harmonic Analysis*, 28(1):104–113, 2010.

[16] Richard Bellman. *Dynamic programming*. Courier Corporation, 2013.

[17] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *NIPS*, pages 2672–2680, 2014.

[18] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. In *NIPS*, pages 2234–2242, 2016.

[19] Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In *NIPS*, pages 2172–2180, 2016.

[20] Hariharan Narayanan and Sanjoy Mitter. Sample complexity of testing the manifold hypothesis. In *NIPS*, pages 1786–1794, 2010.

[21] Martin Arjovsky and Léon Bottou. Towards principled methods for training generative adversarial networks. *arXiv preprint arXiv:1701.04862*, 2017.

[22] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein gan. *stat*, 1050:26, 2017.

[23] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. Improved training of wasserstein gans. In *NIPS*, pages 5767–5777, 2017.

[24] R. Timothy Marler and Jasbir S. Arora. The weighted sum method for multi-objective optimization: new insights. *Structural and Multidisciplinary Optimization*, 41(6):853–862, Jun 2010.

[25] Reinhard Schachtner. *Extensions of non-negative matrix factorization and their application to the analysis of wafer test data*. PhD thesis, 2010.

[26] Tiago J Rato, Jakey Blue, Jacques Pinaton, and Marco S Reis. Translation-invariant multiscale energy-based pca for monitoring batch processes in semiconductor manufacturing. *IEEE Transactions on Automation Science and Engineering*, 14(2):894–904, 2017.

[27] Jong-Min Lee, S Joe Qin, and In-Beum Lee. Fault detection and diagnosis based on modified independent component analysis. *AIChE journal*, 52(10):3501–3514, 2006.

[28] George C Linderman, Manas Rachh, Jeremy G Hoskins, Stefan Steinerberger, and Yuval Kluger. Efficient algorithms for t-distributed stochastic neighborhood embedding. *arXiv preprint arXiv:1712.09005*, 2017.

[29] Peter J Rousseeuw. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics*, 20:53–65, 1987.