Mixture Density Generative Adversarial Networks
Supplementary Material

Hamid Eghbal-zadeh[1] Werner Zellinger[2,3] Gerhard Widmer[1]

[1] LIT AI Lab & Institute of Computational Perception, Johannes Kepler University Linz
[2] Department of Knowledge-Based Mathematical Systems, Johannes Kepler University Linz
[3] Software Compenetce Center Hagenberg GmbH
{hamid.eghbal-zadeh, werner.zellinger, gerhard.widmer}@jku.at

1. Extended Results and Discussion

1.1. Hyper-parameter grid search

We provide evaluation results for hyper-parameter grid search with 3 different variances and 5 different number of gaussian components on two mode discovery datasets (grid 2D in Table 1 and ring 2D in Table 2). The results show that overall the variance of $\frac{1}{4} = 0.25$ achieves better results compared to lower ($\frac{1}{6} = 0.16$) and higher ($\frac{1}{2} = 0.5$) variances. As can be seen, using ery small (0.16) and very large (0.50) variance results in degradation of the results. We can also observe that for datasets with more modes (2D grid), higher number of components improved the results. This suggests that using more components can help samples spread more in the data spaces where more modes are available.

1.2. Grid search plots

We visualize the component assignments in each gaussian during the training for real and fake embeddings. The *X* axis, represents the epochs and the *Y* axis represents the components. Each color in every column shows a different Gaussian component and the width of each color in every column demonstrate the percentage of the embeddings assigned to that gaussian component after one whole epoch. In addition to the component assignments, the probability landscape and the generated samples are also provided for the 2D grid dataset, using 3 different variances of 0.5 in Table 11, 0.25 in Table 12 and variance of 0.16 in Table 13.

1.3. Relation to other GANs

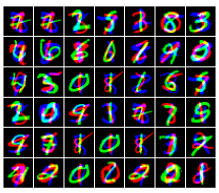
In this section, we review the provided solutions for mode collapse and explain the relation to other GANs that provide solutions for mode collapse.

1.3.1 Auto-encoding for mode discovery

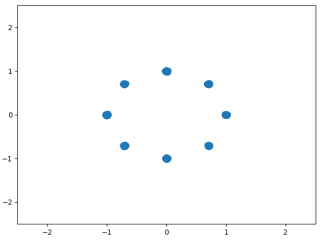
Several GANs including VEEGAN [7] and ALI [2] use an auto-encoding technique as a stabaliser and a method for better mode discovery. Although the authors report better mode discovery properties compared to vanilla GAN, their method require substantially larger number of parameters caused by the additional auto-encoding operations. MD-GAN does not use any auto-encoding or additional networks. We empirically showed that using a single discriminator, and apply the clustering in the discriminator embedding results in significantly better mode discovery properties (discovering more modes, more high quality samples) in various cases.

1.3.2 Additional optimisation steps

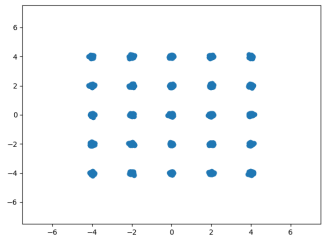
Unrolled GAN [5] proposes to computed several update steps in the generator and use it in the gradient computation of the generator. This way, the generator predicts the future steps of the discriminator and can create better samples resulting in discovering more modes. This solution is computationally very expensive as for each updates in the generator, several updates



(a) Stacked-MNIST.



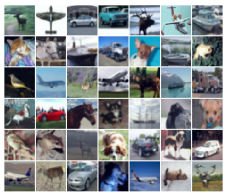
(b) Ring 2D.



(c) Grid 2D.

(d) MNIST.

(e) Fashion-MNIST



(f) CIFAR-10.

(g) CelebA.

Figure 1: Examples of real samples.

of the discriminator (as reported by the authors, 5 updates) have to be computed. MD-GAN does not compute any additional updates, and for every update in the generator, only one update in the discriminator is computed. We also empirically showed that MD-GAN can achieve significantly better mode discovery results compared to Unrolled GAN on several datasets, despite being computationally more efficient.

Table 1: The results of the hyper-parameter grid search for MD-GAN on 2D ring dataset. Every experiment is repeated 5 times.

ngmm	var=0.5		var=0.25		var=0.16	
	modes (8)	% hq	modes (8)	%hq	modes (8)	%hq
4	8.00±0.00	90.85±0.75	7.20±0.40	74.26±9.73	8.00±0.00	89.82±1.32
6	8.00±0.00	92.67±1.9	7.80±0.40	67.71±12.54	8.00±0.00	91.3±1.15
8	8.00±0.00	91.81±2.67	8.00±0.00	95.82±3.71	8.00±0.00	83.52±4.72
10	8.00±0.00	96.48±3.6	8.00±0.00	89.03±3.70	8.00±0.00	93.56±1.13
12	8.00±0.00	95.77±2.04	7.20±0.40	78.87±9.82	6.4±0.80	90.256±2.19

2. Network Architectures

The architectures used for experiments in this paper. Architectures used for MNIST and Fashion-MNIST are provided in Table 5. Architectures used for CIFAR-10 experiments can be found in Table 6 and architectures of CelebA experiments are detailed in Table 7. Architectures of Stacked-MNIST are provided in Table 9 and Table 8. And finally, architectures of Grid 2d and Ring 2D are explained in Table 10.

Scaled Sigmoid is the sigmoid non-linearity with scaled output in [-2.5,2.5]. This choice is based on the limits of our Simplex means which are in the same range.

3. Datasets samples

Samples of real data from all datasets used are provided in Figure 1.

Table 2: The results of the hyper-parameter grid search for MD-GAN on 2D grid dataset. Every experiment is repeated 5 times.

ngmm	var=0.5		var=0.25		var=0.16	
	modes (25)	% hq	modes (25)	%hq	modes (25)	%hq
4	22.67±1.89	67.67±1.94	24.00±0.00	77.55±9.43	16.67±11.79	56.75±40.13
6	25.00±0.00	93.31±0.66	25.00±0.00	87.84±2.55	24.33±0.47	79.81±5.51
8	24.67±0.47	92.04±3.05	25.00±0.00	88.53±5.41	23.67±1.89	76.79±19.14
10	25.00±0.00	93.84±0.00	25.00±0.00	99.36±2.28	24.00±1.41	93.96±0.17
12	25.00±0.00	89.11±5.36	25.00±0.00	93.87±2.28	25.00±0.00	89.21±0.30

Table 3: Hyperparameters used in our mode-collapse experiments on SMNIST dataset after tuning for each model separately. All weights are initialized with $\mathcal{N}(0, \sqrt{1\text{e}{-3}})$. BS: batchsize. Uni.: Uniform. Norm: Normal. NS: Non-Saturating loss [3]. MI: Mutual Information. NCat: dimensionality of categorical (compressible) noise. Sig.: variance penalty. λ: weight for variance penalty. NG: number of Gaussian components (number of Gaussian components in DeliGAN equals the batchsize.). MD: our proposed Mixture Density loss.

method	arch.	lr D	lr G	BS	loss	other	D run	G run	z dim	z dist
SpNorm [6]	$S_{\frac{1}{2}}$	1.5e−4	5e−5	64	hinge	—	2	1	256	Uni.
InfoGAN [1]	$S_{\frac{1}{2}}$	1.5e−4	5e−5	64	NS+MI	NCat=156	1	1	100	Uni.
Deli [4]	$S_{\frac{1}{2}}$	4e−5	3e−5	100	NS+Sig.	NG=100,$\lambda = 0.05$	1	1	256	Norm.
MD-GAN	$S_{\frac{1}{2}}$	1.5e−4	5e−5	64	MD	—	1	1	256	Uni.
SpNorm [6]	$S_{\frac{1}{4}}$	1.5e−4	5e−5	64	hinge	—	2	1	256	Uni.
InfoGAN [1]	$S_{\frac{1}{4}}$	5e−5	5e−5	64	NS+MI	NCat=156	1	1	100	Uni.
Deli [4]	$S_{\frac{1}{4}}$	4e−5	3e−5	100	NS+Sig.	NG=100,$\lambda = 0.05$	1	1	256	Norm.
MD-GAN	$S_{\frac{1}{4}}$	1.5e−4	5e−5	64	MD	—	1	1	256	Uni.

Table 4: Hyperparameters used in our mode-collapse experiments on 2D datasets after tuning for each model separately. All weights are initialized with $\mathcal{N}(0, 0.02)$. BS: batchsize. Uni.: Uniform. Norm: Normal. NS: Non-Saturating loss [3]. MI: Mutual Information. NCat: dimensionality of categorical (compressible) noise. Sig.: variance penalty. λ: weight for variance penalty. NG: number of Gaussian components. MD: our proposed Mixture Density loss.

method	lr D	lr G	BS	loss	other	D run	G run	z dim	z dist
SpNorm [6]	1e−3	1e−3	500	hinge	—	1	1	2	Uni.
InfoGAN [1]	1e−3	1e−3	500	NS+MI	NCat=2	1	1	2	Uni.
Deli [4]	1e−3	1e−3	500	NS+Sig.	NG=500,$\lambda = 0.05$	1	1	2	Norm.
MD-GAN	1e−3	1e−3	500	MD	—	1	1	2	Uni.

Table 5: The architectures used in MD-GAN for MNIST and Fashion-MNIST experiments. The dimensionality of the simplex is d.

Discriminator	Generator
Input $1 \times 28 \times 28$ gray-scale	Input 1×100
Conv (64, 4×4, stride=2, LReLu)	FC (1024, ReLu) + BN
Conv (128, 4×4, stride=2, LReLu) + BN	FC ($7 \times 7 \times 128$, ReLu) + BN
FC (128, LReLu) + BN	UpConv(64, 4×4,stride=2,RELU)+ BN
FC (d , ScaledSigmoid(-2.5,2.5))	UpConv(1, 4×4,tanh)

References

[1] Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In *Advances in Neural Information Processing Systems*, 2016. 3

Table 6: The architectures used in MD-GAN for CIFAR-10 experiments. The dimensionality of the simplex is d.

Discriminator	Generator
Input $3 \times 32 \times 32$ RGB	Input 1×100
Conv (64, 4×4, stride=2, LReLu)	FC ($2 \times 2 \times 448$, ReLu)+BN
Conv (128, 4×4, stride=2, LReLu) + BN	UpConv(256, 4×4,stride=2,RELU)+ BN
Conv (256, 4×4, stride=2, LReLu) + BN	UpConv(128, 4×4,stride=2,RELU)
FC (128, LReLu) + BN	UpConv(64, 4×4,stride=2,RELU)
FC (d , ScaledSigmoid(-2.5,2.5))	UpConv(3, 4×4,tanh)

Table 7: The architectures used in MD-GAN for CelebA experiments. The dimensionality of the simplex is d.

Discriminator	Generator
Input $3 \times 64 \times 64$ RGB	Input 1×100
Conv (64, 4×4, stride=2, LReLu)	FC ($4 \times 4 \times 128$, ReLu)+BN
Conv (128, 4×4, stride=2, LReLu) + BN	UpConv(256, 4×4,stride=2,ReLu)+ BN
Conv (256, 4×4, stride=2, LReLu) + BN	UpConv(128, 4×4,stride=2,ReLu)
FC (128, LReLu) + BN	UpConv(64, 4×4,stride=2,ReLu)
FC (d , ScaledSigmoid(-2.5,2.5))	UpConv(3, 4×4,tanh)

Table 8: The architecture **B** used in MD-GAN for Stacked-MNIST. The dimensionality of the simplex is d.

Discriminator	Generator
Input $3 \times 28 \times 28$ gray-scale	Input 1×256
	FC ($7 \times 7 \times 128$, ReLu) + BN
Conv (64, 4×4, stride=2, LReLu)	UpConv(256, 4×4,stride=2,ReLu)+ BN
Conv (128, 4×4, stride=2, LReLu) + BN	UpConv(128, 4×4,stride=2,ReLu)
Conv (256, 4×4, stride=2, LReLu) + BN	UpConv(64, 4×4,stride=1,ReLu)
FC (d , ScaledSigmoid(-2.5,2.5))	UpConv(3, 4×4,stride=1,tanh)

Table 9: The architecture S_X used in MD-GAN for Stacked-MNIST. The dimensionality of the simplex is d. X is the amount of parameter reduction ($\frac{1}{2}$ or $\frac{1}{4}$).

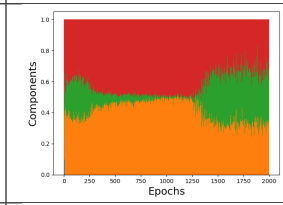
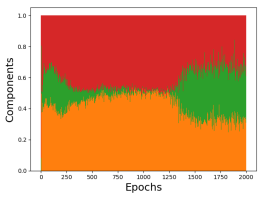
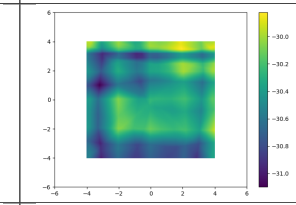
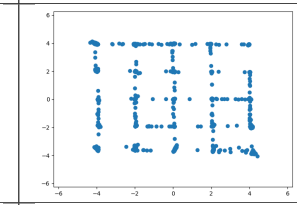
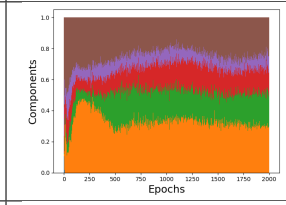
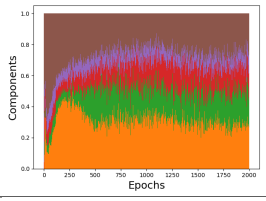
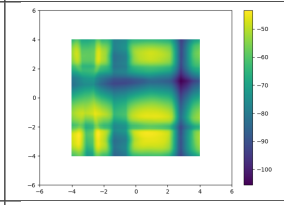
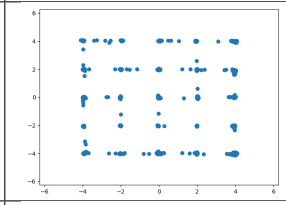
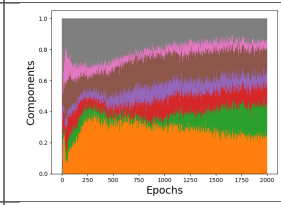
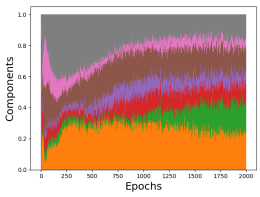
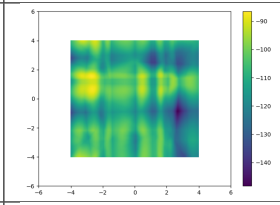
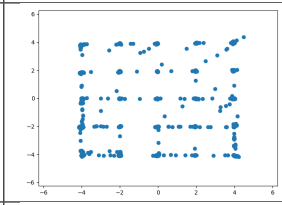
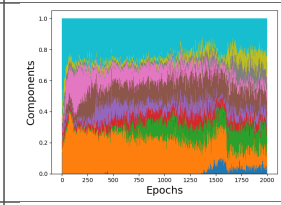
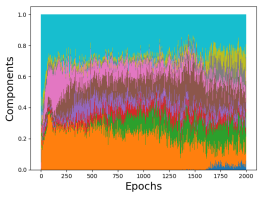
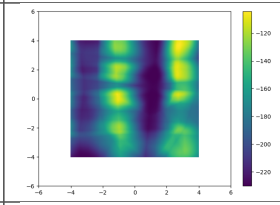
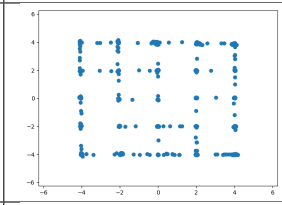
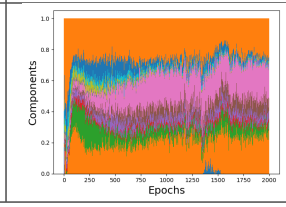
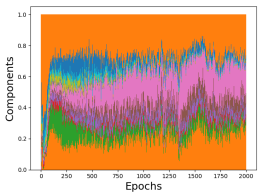
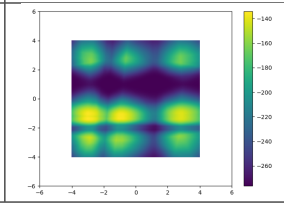
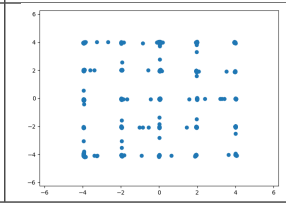
Discriminator	Generator
Input $3 \times 28 \times 28$ gray-scale	Input 1×256
	FC ($4 \times 4 \times 512 \times X$, ReLu) + BN
Conv ($64 \times X$, 3×4, stride=2, LReLu)	UpConv($256 \times X$, 4×4,stride=2,RELU)+ BN
Conv ($128 \times X$, 4×3, stride=2, LReLu) + BN	UpConv($128 \times X$, 4×4,stride=2,RELU)+ BN
Conv ($256 \times X$, 4×3, stride=2, LReLu) + BN	UpConv($64 \times X$, 4×4,stride=2,RELU)+ BN
FC (d , ScaledSigmoid(-2.5,2.5))	UpConv(3, 4×4,tanh)

Table 10: The architectures used in MD-GAN for synthetic data experiments. The dimensionality of the simplex is d. The discriminator's output and dimensionality is changed to the ones used in their original paper.

Discriminator	Generator
Input 2 FC(128)-LReLU	Input 2 FC(128)-ReLU
FC(128)-LReLU	FC(128)-ReLU
FC(d)-ScaledSigmoid(-2.5,2.5)	FC(2)-ScaledTanh(-6,6)

[2] Vincent Dumoulin, Ishmael Belghazi, Ben Poole, Alex Lamb, Martin Arjovsky, Olivier Mastropietro, and Aaron Courville. Adversarially learned inference. *arXiv preprint arXiv:1606.00704*, 2016. 1

Table 11: Real and fake cluster assignments, probability landscape and generated samples in MD-GAN for grid-2D with different number of components (# in first column) and Variance of 0.5.

#	Real cluster assignments	Fake cluster assignments	Probability landscape	Generated samples
4				
6				
8				
10				
12				

[3] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, 2014. 3

[4] Swaminathan Gurumurthy, Ravi Kiran Sarvadevabhatla, and R Venkatesh Babu. Deligan: Generative adversarial networks for diverse and limited data. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2017. 3

[5] Luke Metz, Ben Poole, David Pfau, and Jascha Sohl-Dickstein. Unrolled generative adversarial networks. *International Conference on Learning Representations*, 2017. 1

[6] Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. *International Conference on Learning Representations*, 2018. 3

[7] Akash Srivastava, Lazar Valkoz, Chris Russell, Michael U Gutmann, and Charles Sutton. VEEGAN: Reducing mode collapse in gans using implicit variational learning. In *Advances in Neural Information Processing Systems*, 2017. 1

Table 12: Real and fake cluster assignments, probability landscape and generated samples in MD-GAN for grid-2D with different number of components (# in first column) and Variance of 0.25

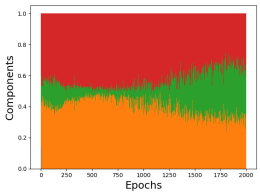
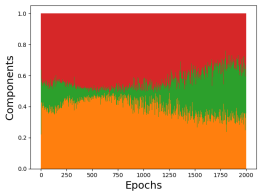
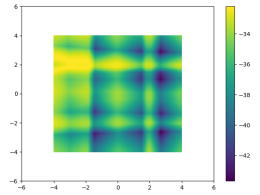
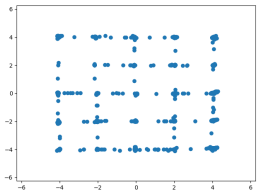
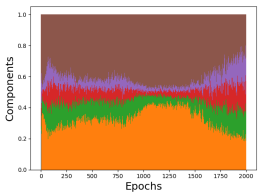
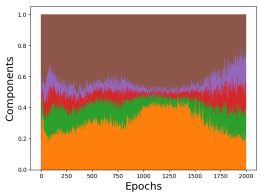
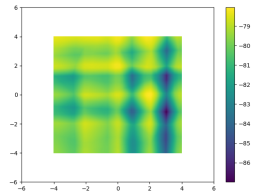
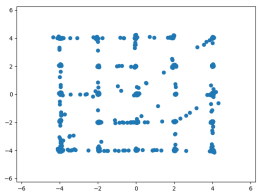
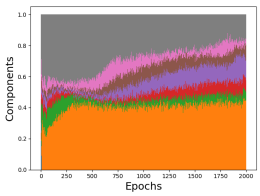
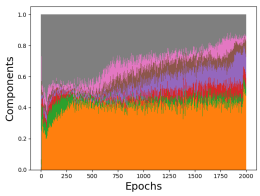
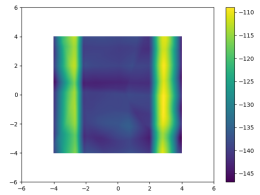
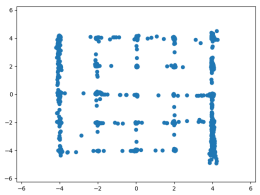
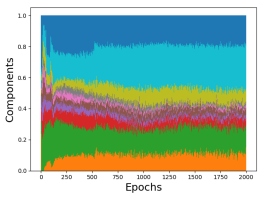
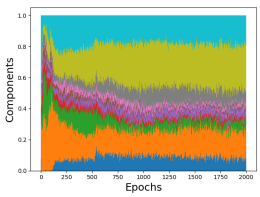
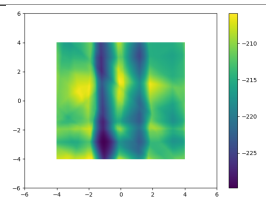
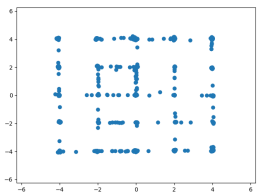
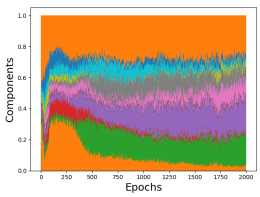
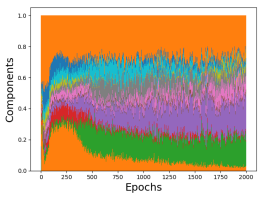
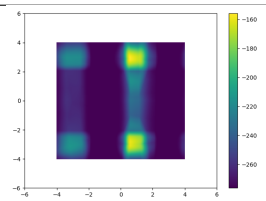
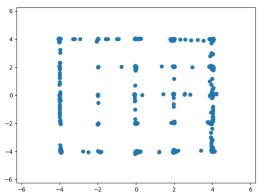
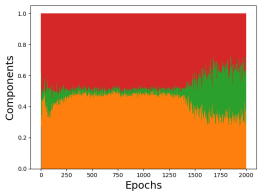
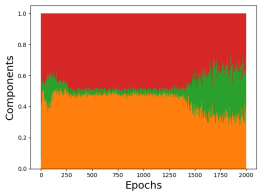
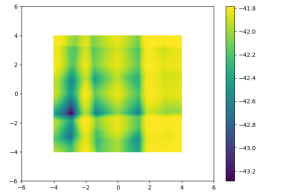
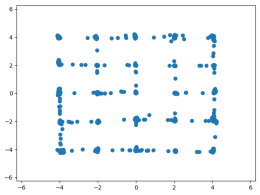
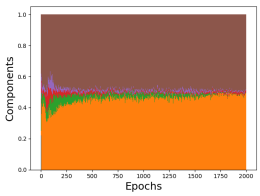
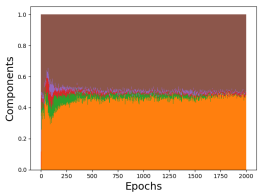
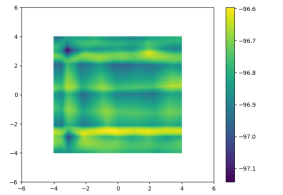
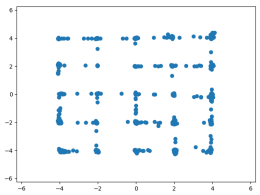
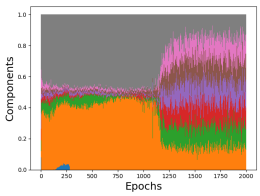
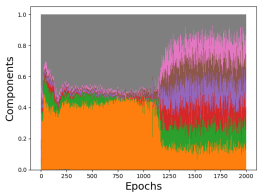
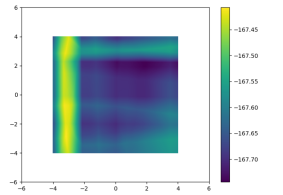
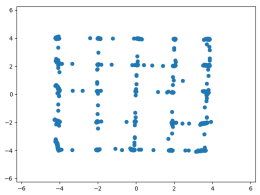
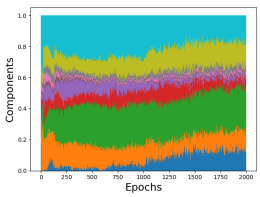
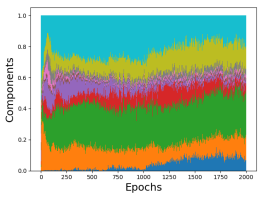
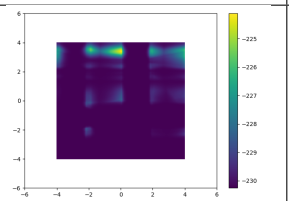
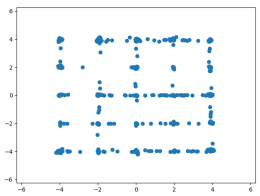
#	Real cluster assignments	Fake cluster assignments	Probability landscape	Generated samples
4				
6				
8				
10				
12				

Table 13: Real and fake cluster assignments, probability landscape and generated samples in MD-GAN for grid-2D with different number of components (# in first column) and Variance of 0.16

#	Real cluster assignments	Fake cluster assignments	Probability landscape	Generated samples
4				
6				
8				
10				
12