IMPROVING DIVERSITY IN GENERATIVE ADVERSARIAL NETWORKS BY ENCOURAGING DISCRIMINATOR REPRESENTATION ENTROPY

Anonymous authors
Paper under double-blind review

ABSTRACT

We propose a novel regularizer for the training of Generative Adversarial Networks (GANs) so that the generator can better capture the variations within the real data distribution, thus helps to avoid subtle model collapse and improve the performance of GANs. The idea is to encourage the discriminator D to provides more informative signals for the learning of the generator G by allocating the model capacity of D in a more desirable way. In particular, we measure the model capacity of D by its activation patterns, and our new regularizer is constructed to encourage a high joint entropy of the activation patterns of the hidden layers of D. Experimental results on both synthetic data and real datasets show that our regularizer helps to improve the sample quality in the unsupervised learning setting, and also the classification accuracy in the semi-supervised learning setting.

1 INTRODUCTION

Generative Adversarial Networks (GANs) (Goodfellow et al., 2014) learn from unlabeled data by engaging the generative model (G) in an adversarial game with a discriminator (D). GANs provide a promising new approach to unsupervised learning of complex high dimensional data, with successful applications on image data (Isola et al., 2016; Shrivastava et al., 2016), and great potential for predictive representation learning (Mathieu et al., 2015) as well as reinforcement learning (Finn et al., 2016; Henderson et al., 2017).

However, challenges in GANs training impede their wider applications in new data domain and as building blocks in other models. Many architectures and techniques have been proposed (Radford et al., 2015; Salimans et al., 2016; Gulrajani et al., 2017) to reduce extreme failures and improve realism of generated data. With the various techniques, practitioners can often train GAN without extreme collapse and heavily oscillation nowadays on standard datasets. However, a number of theoretical and practical open problems still remain. In particular, G often fails to capture certain variation in data or modes of real data distribution, while D fails to exploit the failure to provide better training signal to G , leading to subtle mode collapse. Arora et al. (2017) shows that with insufficient capacity, D could fail to distinguish real and generated data distributions even when they are their Jensen-Shannon divergence or Wasserstein distance is not small. However Arora et al. (2017)'s capacity notion is a static one. During training D 's finite capacity allocation in input space is changing. A more refined and more dynamic analysis of capacity is desired.

In this work, we demonstrate that even with sufficient maximum capacity, D might not allocate it in a way that helps GAN training reaching equilibrium. We then propose a novel regularizer that to encourage D 's hidden binary activation to have high joint entropy. Our experiments show that high entropy representation leads to improved sample quality and semi-supervise learning results.

Several regularization/gradient penalty techniques have recently been proposed to stablize GANs training (Gulrajani et al., 2017; Mescheder et al., 2017). Since we are motivated from the capacity angle, our method differs from them the most in that we regularize D's hidden layer acitivities, neither on input gradients nor parameters directly. This in turn will constraint the parameters of D and hence G. In terms of functional forms, (Gulrajani et al., 2017) adds gradient penality to the discriminator D's input gradient, $\nabla_x D(x)$, to indirectly regularize D and G's weights. Both

(Mescheder et al., 2017) regularize the training dynamics by constraining the parameters directly via adding other vector fields to the gradient updates.

2 RECTIFIER NET CAPACITY USAGE AND EFFECTS ON GAN

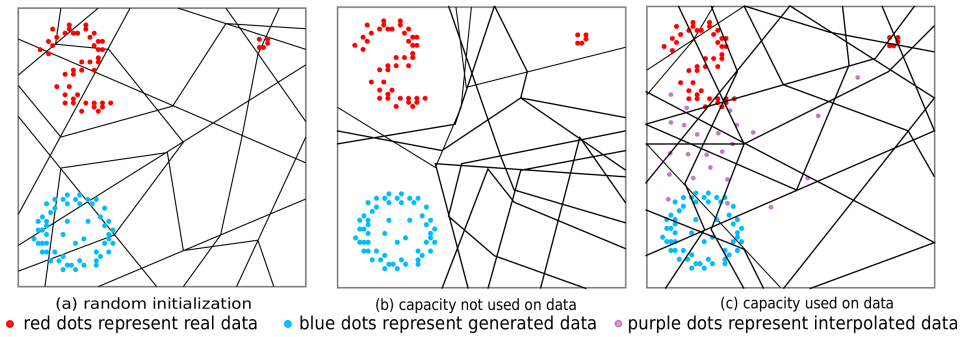

Figure 1: Capacity usage in different scenarios. D (a rectifier net) cuts the input space into convex polytopes, and on each region it is a linear function. **left**: D uniformly spreads its capacity in the input space, but does not have enough capacity to distinguish all subtle variation within a data distribution. **middle**: D uses its capacity in the region with no data; while real and fake data are correctly seperated, variations within real data distribution are not represented by D , so cannot possibly be communicated to G if this degeneracy persists through training; meanwhile all fake points in the same linear region passes the same gradient information to G , even if they are visually distinct. **right**: D spends most capacity on real and fake data, but also in region where G might move its mass to in future iterations.

The motivation of our proposed regularizer starts with an observation that during the training of GANs, the generator G only receives information about real data distribution and its implicit manifold indirectly through the gradient of D, $\nabla_x D(x)$. However, the task of the discriminator D is solely to separate real samples from the generated fake ones. When real and fake data are far apart in the early stages of the training or when the generator G collapses into some particular modes, the discriminator D has no incentive to allocate its capacity in the most desired way, and thus it may only provides very limited information for G, incapable to guide the generator G to capture different variations within the real data distribution. See Figure 1 for a pictorial illustration. In this paper we propose a new regularizer to encourage the discriminator D to allocate its model capacity in a more desirable way to provides extra guidance for the learning of G. In particular, we would like the discriminator D to allocate its model capacity to tell apart the generated fake samples from the real ones in *distinct* ways, and not waste the capacity on the regions where there is no data.

Typically, the discriminator D uses rectifier as its activation function. As illustrated in Figure 1, the discriminator D essentially cuts the input space into different local linear regions. Intuitively, for a fixed region in the input space, how much capacity the discriminator D put into can be approximately measured by the number of local linear pieces in the region(Montufar et al., 2014). The feedback information for the generator G is essentially the slope of the corresponding local linear region defined by D. As the generated data points x lie in different regions, $\nabla_x D(x)$ are likely to be diverse in different directions. As such the generator G in turn can receive diverse learning signals from $\nabla_x D(x)$. Further note that the different local linear regions defined by D is closely related to the different activation patterns of D (Raghu et al., 2016). More specifically, two generated data points with different activation patterns in D are guaranteed to lie on different linear regions. In this paper we propose a regularizer to encourage the diversity of the activation patterns of D, in order to guide the discriminator D to allocate its model capacity in such way that G can receive more informative signals.

The rest of the paper is organized as follows. Related works in the literature are discussed in Section 2.1. We formally present our regularizer in Section 3. The proposed regularizer will be applied on fake data (blue in Fig. 1) and region inbetween real and fake data (purple in 1), together with auto-encoding or other auxiliary tasks on the real data forms our proposed strategy for improving GAN training.

2.1 RELATED WORKS

The role of model capacity of the discriminator D in training generative adversarial networks (GANs) has been previously explored by Arora et al. (2017). While they show that a discriminator D with finite number of parameters has a limited capacity in distinguishing real data and the generator G's samples (and suggest to increase the discriminator D's capacity), the question on *where* and *how* D can utilize its limited capacity effectively is left unanswered. Our paper can be seen as a work continuing along this line, providing guidance for better usage of the model capacity.

Encouraging D to use its capacity in a constructive way is non-trivial. Auto-encoding methods appear promising, and have been explored in Energy Based GAN (Zhao et al., 2016) and Boundary Equilibrium GAN (Berthelot et al., 2017). In their work, energy is pushed down on real data manifolds while energy is pushed up for generated data. Because their framework involves auto-encoding tasks, they naturally require D to learn almost all essential features (v.s. plain classification tasks). However, when fake data is closer to real ones, there appears to be less-understood tradeoffs in sample quality and sample diversity where Berthelot et al. empirically observe. Some mechanisms potentially hinder how well D can separate real from generated data while capturing different variations within the real data distribution.

Another potential way to mitigate the problem is to use a Bayesian neural net, whose model capacity away from data is not degenerate. However, computationally scalable deep Bayesian neural networks are still an active area of research **?** and are not easy to use. Alternatively, we can use auxiliary tasks to regularize D 's capacity usage. On real data manifold, this is generally easy. If the data has label, augmenting D with semi-supervised learning task, as shown in **?**, improves GAN training stability and the resulting generative model. If the data domain has other structures, for instance order or context structure in many sequence data, they can be exploited to regularize the model capacity usage as well and should in principle improve GAN training. Finally, if there is no label, we can use auto-encoding as an auxiliary task on the real data distribution.

3 BINARIZED REPRESENTATION ENTROPY

We now introduce a regularizer that encourages the hidden representation of different data points to be diverse. The regularizer acts on the binarized activation pattern, and has two terms: the first term, R_{ME}, encourages individual hidden units to be on half of the time on average, to have high *marginal entropy*; the second term, R_{AC}, encourages pairs of hidden units to have low *activation correlation*. After defining the terms mathematically below, we will first motivate their forms as a necessary condition for maximizing the representation entropy in Sec. 3.2. Then we will show that minimizing the proposed regularizer leads to a lower bound on the joint entropy of the binarized representation in Sec. 3.3.

3.1 THE BINARY REPRESENTATION ENTROPY REGULARIZER

We present the formal definition of the binary representation entropy regularizer in this section. Given a mini-batch, $\{x_1, \ldots, x_K\}$ of size K, let $\mathbf{h}_i \in \mathbb{R}^d$ be the immediate pre-nonlinearity activity of data point x_i in a mini-batch on a particular layer of d hidden units. $\mathbf{h}_i$ is represented as a column vector. Let $\mathbf{s}_i = \text{sign}(\mathbf{h}_i) := \frac{\mathbf{h}_i}{|\mathbf{h}_i|} \in \{\pm 1\}^d$ be the sign function, where $|\cdot|$ is entry-wise absolute value. Assume that the sign function $\mathbf{s}_i$ of each data point x_i is an independent sample of $(U_1, \ldots, U_d)$, where $(U_1, \ldots, U_d)$ denote a multivariate Bernourlli random vector with parameters $(p_1, \ldots, p_d)$. Also denotes its joint distribution function by $\mathbb{P}$ and its kth marginal Bernoulli distribution function by $\mathbb{P}_k$. We would like to construct a regularizer that encourages $\mathbb{P}$ to have larger entropy. Ideally, one could use an empirical estimate of the entropy function as a desired regularizer. However, estimating the entropy of a high-dimensional distribution based a small number of samples has been well known to be difficult, especially with a small mini-batch size (Darbellay & Vajda, 1999; Miller, 2003; Kybic, 2007; Kybic & Vnučko, 2012; Scott, 2015).

In this paper, we propose a simple regularizer R_{BRE} which, as shown below, encourages the large entropy of $\mathbb{P}$ in a weak manner. Our proposed *Binary Representation Entropy* (BRE) regularizer R_{BRE} applied on a given layer consists of two terms defined on $\{\mathbf{s}_1, \ldots, \mathbf{s}_K\}$: $R_{\text{BRE}} = R_{\text{ME}} + R_{\text{AC}}$,

where

$$R_{\text{ME}} = \frac{1}{d}\sum_{k=1}^{d} \bar{\mathbf{s}}_{(k)}^2; \quad \text{and} \quad R_{\text{AC}} = \text{avg}_{i\neq j}|\mathbf{s}_i^\top \mathbf{s}_j|/d.$$

Here $\bar{\mathbf{s}}_{(k)}$ is the average of the sign function on the kth hidden unit across the mini-batch, and $\text{avg}_{i\neq j}$ denotes the average taken over all possible pairs of (i, j) such that $i \neq j$. Thus R_{ME} can be interpreted as an empirical estimate of $\frac{1}{d}\sum_{k=1}^{d} \mathbb{E}\left[U_k\right]^2$, and R_{AC} as an empirical estimate of $\mathbb{E}\left[\,|\sum_{k=1}^{d} U_k\tilde{U}_k|\,\right]$, where $\tilde{U}_k$ is a random sample of $\mathbb{P}_k$ that is independent to U_k. It is straightforward that the maximum entropy of $\mathbb{P}$ is achieved when $p_k = 1/2$ and $(U_1, \ldots, U_d)$ are mutually independent. Intuitively, R_{ME} encourages p_k to be close to $1/2$, while R_{AC}, as shown in Section 3.3, encourages $(U_1, \ldots, U_d)$ to be pairwise independent. Although pairwise independence is weaker than mutual independence, we will see in Section 3.3 that pairwise independence guarantees a lower bound for the entropy of the joint distribution $\mathbb{P}$.

In practice, due to the degenerate gradient of the sign function, we replace $\mathbf{s}_i$ in R_{ME} by its smooth approximation $\mathbf{a}_i = \text{softsign}(\mathbf{h}_i) := \frac{\mathbf{h}_i}{|\mathbf{h}_i|+\epsilon}$. We also relax R_{AC} by allowing a soft margin term, as $R_{\text{AC}} = \text{avg}_{i\neq j} \max\left(0, |\mathbf{a}_i^\top \mathbf{a}_j|/d - \eta\right)$, where $\eta = 3\sqrt{1/d}$. We defer discussing the choice of ϵ to the end of this section.

3.2 Maximum entropy representation has $R_{\text{BRE}} \approx 0$.

In this section we show that $R_{\text{BRE}} \approx 0$ is a necessary condition for $\mathbb{P}$ to achieve maximum entropy. The binarized representation on a layer attain its maximal entropy, when U_k's are mutually independent and $\mathbb{E}\left[U_k\right] = 0$, for all $k \in \{1, ..., K\}$. Therefore, the average signed activation $\bar{\mathbf{s}}_k$ is approximately zero, and so is R_{ME}.

For R_{AC}, lets consider $\sum_{k=1}^{d} M_k$, where $M_k = U_k\tilde{U}_k$. Note that given $\mathbb{P}_k$ being Bernoulli(0.5) and U_k's are mutually independent, M_k's are mutually independent and have the distribution of Bernoulli(0.5) as well. Therefore by the Central Limit Theorem, the distribution of $\sum_{k=1}^{d} M_k$ converges in distribution to the Gaussian distribution $N(0, d)$. Therefore, given sufficiently large d, the distribution of $\frac{\sum_{k=1}^{d} M_k}{d} = \frac{\mathbf{s}_i^\top \mathbf{s}_j}{d}$ is approximately $N(0, 1/d)$. A good choice for the margin threshold is $\eta = c\sqrt{1/d}$, where we choose $c = 3$ to leave 99.7% of i, j pairs unpenalized in the maximum entropy case. [1]

3.3 Minimizing the BRE regularizer to the entropy of multivariate Bernoulli distribution

We further show that minimizing R_{BRE} is equivalent to enforcing the maximum entropy of each $\mathbb{P}_k$ and the pairwise independence of $(U_1, \ldots, U_d)$. To see that, first note that each summand $\bar{\mathbf{s}}_k$ in R_{ME} is the empirical estimate of $2p_k - 1$, the mean of the marginal distribution $\mathbb{P}_k$. Thus minimizing $\bar{\mathbf{s}}_k^2$ leads to $p_k = \frac{1}{2}$, i.e. maximizing the entropy of $\mathbb{P}_k$. Moreover, the regularizer R_{AC} is essentially equivalent to minimizing $\left(\mathbf{s}_i^\top \mathbf{s}_j\right)^2$. The following proposition shows that given U_k being zero-mean, R_{AC} tends to enforce $\text{Cov}\left(U_k, U_t\right) = 0$, i.e. U_k's are pairwise independent.

Proposition 3.1. *Let $U = (U_1, \ldots, U_d)$ be a zero-mean multivariate Bernoulli vector of $\mathbb{P}$, and $\tilde{U} = (\tilde{U}_1, \ldots, \tilde{U}_d)$ denotes another random vector of $\mathbb{P}$ that is independent to U. Then*

$$\mathbb{E}\left[\left(U\tilde{U}\right)^2\right] = \mathbb{E}\left[\left(\sum_{k=1}^{d} U_k\tilde{U}_k\right)^2\right] = d + \sum_{i\neq j} Cov\left(U_k, U_j\right)^2.$$

We prove this proposition in Appendix A.

Lastly, Assuming the hidden units U_k's are zero-mean and pairwise independent, by Corollary 3.3 of Gavinsky & Pudlák (2015) (which we restate here for completeness), the entropy of $\mathbb{P}$ satisfies

$$H(\mathbb{P}) \geq \log(n + 1).$$

[1] Note that the expectation of R_{AC} under the maximum entropy assumption is not zero, but a small number on the order of $1e - 3$.

Theorem 3.2 (Corollary 3.3 of Gavinsky & Pudlák (2015)). *Let* $H_{\min}(\mathbb{P}) = -\log\left(\max_x \mathbb{P}(X = x)\right)$. *Also let* $(U_1, \ldots, U_d)$ *be pairwise independent random variable of Bernoulli(*0.5*). Then,*

$$H(\mathbb{P}) \geq H_{\min}(\mathbb{P}) \geq \log(n+1).$$

3.4 CONTROLLING THE SOFTNESS OF SOFT-SIGN APPROXIMATION

R_{BRE} is defined on $a_k = \text{softsign}(h_k) := \frac{h_k}{|h_k|+\epsilon}$, so ϵ is a hyperparameter to be chosen. If ϵ is too small, the nonlinearity becomes too non-smooth for stochastic gradient descent; if it is too large, it fails to be a good approximation to the sign function for R_{BRE} to be useful. Furthermore, not only different layers could have different scales of h_k, hence requiring different ϵ, during training the scale of $\mathbf{h}$ could change too. Therefore, instead of setting a fixed ϵ, we set $\epsilon = s\overline{|h_k|}$, where $\overline{|h_k|}$ averages over the d dimensions of the layer and s is a constant with value 0.001. In this way, softsign$(\cdot)$ is invariant with respect to any multiplicative scaling of h_k in the forward pass of the computation; in the backward pass for the gradient computation, we do not backpropagate through ϵ. We choose $s = 0.001$, as we observe empirically that this usually makes 90% to 99% of units to have absolute value at least .9.

3.5 WHY REGULARIZING $\nabla_x D(x)$ DIVERSITY WOULD NOT WORK?

Because G receives training signal from D , so it might be tempting to add a regularizer to enforce gradient directions $\nabla_x D(x_i)_i$ to be diverse. However, in rectifier networks, if two inputs share the same activation pattern, the input gradients located at the two points are co-linear, hence any gradient based learning with such diversity regularizer would have difficulty pulling them apart. In general, unlike R_{BRE} which operates directly on both activated and non-activated path, input gradient regularizer can only access information on the activated path in the network, so that it can only encourage exisiting non-shared activated path, but cannot directly create new non-shared activated path. In theory, other nonlinearities such as tanh might allow input gradient regularizer to work, but they are very hard to train at the the first place. In our preliminary experiments, on tanh nonlinearity networks, input gradient diversity regularizer with either cosine similarity or soft-sign based regularizer like R_{BRE} does not work.

3.6 GAN TRAINING WITH BRE REGULARIZER

WHERE IS R_{BRE} APPLIED?

To regularize GAN training, R_{BRE} is applied to the immediate pre-nonlinearity activities on selected layers of D . Therefore, if there is any normalization layer before nonlinearity, R_{BRE} needs to be applied after the normalization. We emphasize that we use soft-sign for the regularizer only, we do not modify the nonlinearity or any other structure of the neural net.

WHICH LAYERS SHOULD R_{BRE} BE APPLIED ON?

Technically R_{BRE} can be applied on any rectifier layer before the nonlinearity. However, having it on the last hidden rectifier layer before classification might hinder D 's ability to separate real from fake, as the high entropy representation encouraged by R_{BRE} might not be compatible with linear classification. Therefore, for unsupervised learning, we apply R_{BRE} on all except the last rectifier nonlinearity before the final classification; for semi-supervised tasks using the $K+1$-way classification setup from Salimans et al. (2016), we apply R_{BRE} only on 2nd, 4th and 6th convolutional layer, leaving 3 nonlinear layers un-regularized before the final softmax.

WHICH PART OF THE DATA SHOULD R_{BRE} BE APPLIED ON?

Recall from Sec. 2 that we want D to spend enough capacity on both the real data manifold, and the current generated data manifold by G , as well as having adequate capacity on regions where we do not currently observe real or fake points but might in future iterations. To enforce this, we apply R_{BRE} on selected layers on generated data minibatch, as well as random interpolation inbetween real and generated data. Specifically, let $\mathbf{x}_i$ and $\tilde{\mathbf{x}}_i$ be a real and a fake data points respectively,

we sample $\alpha_i \sim U(0,1)$ and let $\hat{\mathbf{x}}_i = \alpha_i \mathbf{x}_i + (1-\alpha_i)\tilde{\mathbf{x}}_i$, and apply R_{BRE} on selected layer representation computed on interpolated data points $\{\hat{\mathbf{x}}_i\}$ as well.

4 EXPERIMENTS

4.1 SYNTHETIC DATASET: MIXTURE OF GAUSSIANS

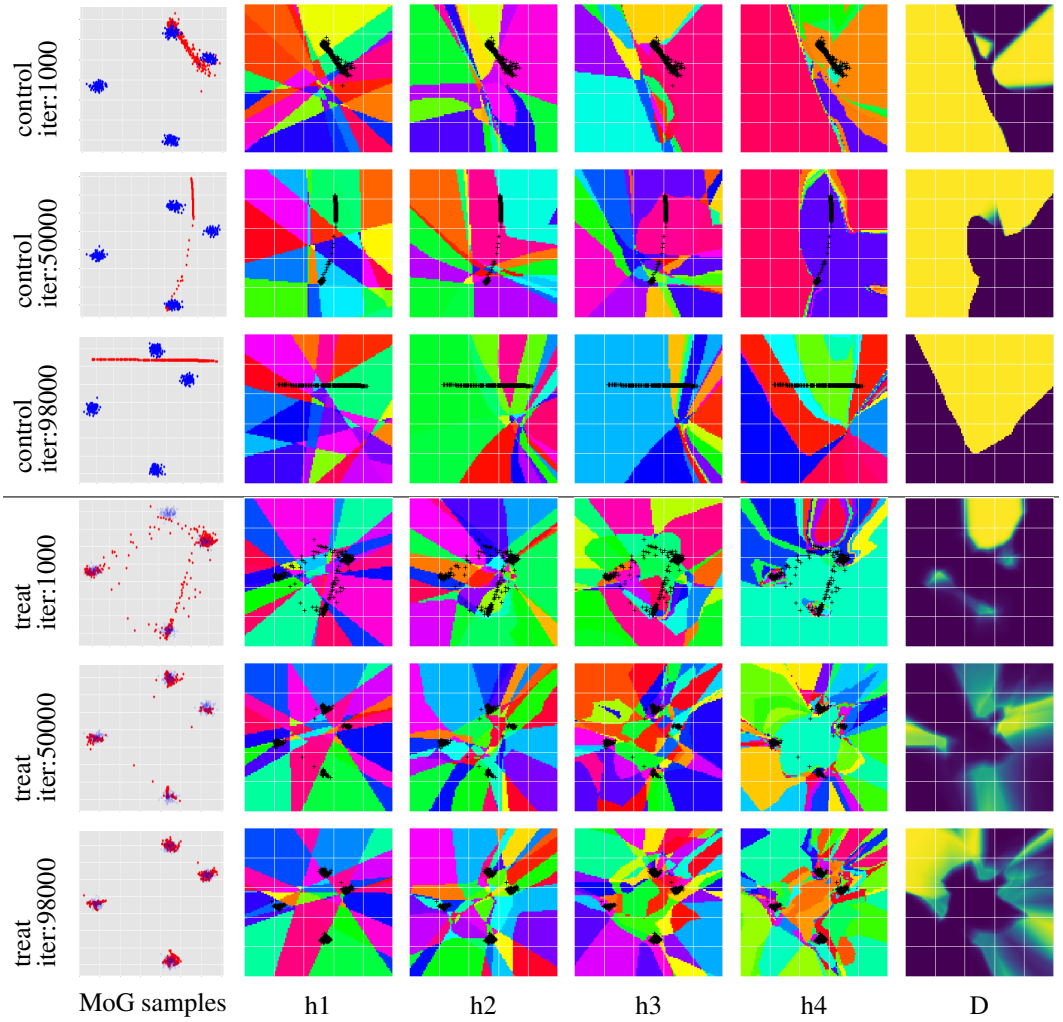

Figure 2: Fitting 2D Mixture of Gaussian (MoG): $h1$-$h4$ show the input space linear region defined by different binary activation patterns on each layer, where each colour corresponds to one unique binary pattern; the last column shows probability of real data accoring to D ; BRE regularizer is applied on $h2$. G is 4-layer (excluding noise layer) MLP with ReLU hidden activation function, and tanh visible activation; D is 5-layer MLP with LeakyRelu(.2). Both D and G has 10 units on each hidden layer, no batch or other normalization is used in D or G , no other stabilization techniques are used; lr=.002 with adam(.5, .999), and BRE regularizer weight 1.; both lr and BRE weight linearly decay to over iterations to $1e-6$ and 0 respectively.

We first demonstrate BRE regularizer's effect on fitting 2D mixture of Gaussian. The top half of Figure 2 contains results from a control experiment (no BRE regularizer). The bottom half of the figure contains results regularized by BRE. Within each setting, each row represents one iteration. We show the beginning, middle, and the end of the training process. The first column shows real data points (blue) and generated data points (red). The second to fifth columns show hidden layers 1 to 4 of D . Pixels with the same color has the same binary activation pattern on that particular layer.

The last column shows the probability of real data according to D . The BRE regularizer is added on $h2$. We show more results at Appendix B

We can see that by adding BRE, in h2, h3, and h4, the input domain is divided into more pieces. This indicates the better usage of D's model capacity, and resulted in better exploration of the space and more stable training.

4.2 CELEBA

We compare stable and unstable runs of DCGAN on CelebA dataset (Liu et al., 2015), as well as effect of BRE regularizer. Fig. 3(a) shows thresholded R_{AC} term through training. The model being investigated is a 4-layer DCGAN for both G and D , with batch normalization. The unstable run (Fig. 3(b)) uses a large initial learning rate of .01 and 3 D update steps for each G update, whereas the stable run (Fig. 3(d)) uses initial $lr = 2e - 3$ and 1 D update for each G update. Even without the BRE regularizer, we can see that when GAN training is stable, D uses more capacity around fake and real data as well as inbetween, as measured by R_{AC} values in Fig. 3(a). When BRE regularizer is applied, the usage of D 's capacity is more improved, and resulted in more diversity in the learned distribution by G (Fig. 3(c)).

Figure 3: (Thresholded) Activity correlation (AC) values (top left) and samples at iteration 10K: (top right) DCGAN unstable run ($lr = .01$ and 3 D update steps for each G update); (lower right) DCGAN stable run ($lr = 2e{-}3$ and 1 D update steps for each G update); (lower left) BRE-DCGAN, DCGAN training with BRE regularizer same hyperparameters as DCGAN stable run in lower right plot. BRE-DCGAN results are visibly more diverse.

4.3 QUANTITATIVE RESULTS ON CIFAR10: IMPROVED INCEPTION SCORE IN UNSUPERVISED SETTING AND HIGHER SEMI-SUPERVISED CLASSIFICATION ACCURACY

First, we demonstrate quantitatively that BRE regularization improves learned distribution as measured by Inception score (Salimans et al., 2016) on CIFAR10 dataset (Krizhevsky, 2009). Fig. 4 shows inception score and (thresholded) activation correlation values (R_{AC}) during training iteration, for two different experiments with a normal optimization setting (4 left column) and a more aggressive optimization setting (4 right column). In both cases, BRE regularization keeps activation correlation low, leading to higher representation entropy by D , and the resulting in higher inception scores. In Table. 1 in Appendix, we show more quantitative results as vary the size of D , and demonstrate that BRE regularization consistently improves inception score.

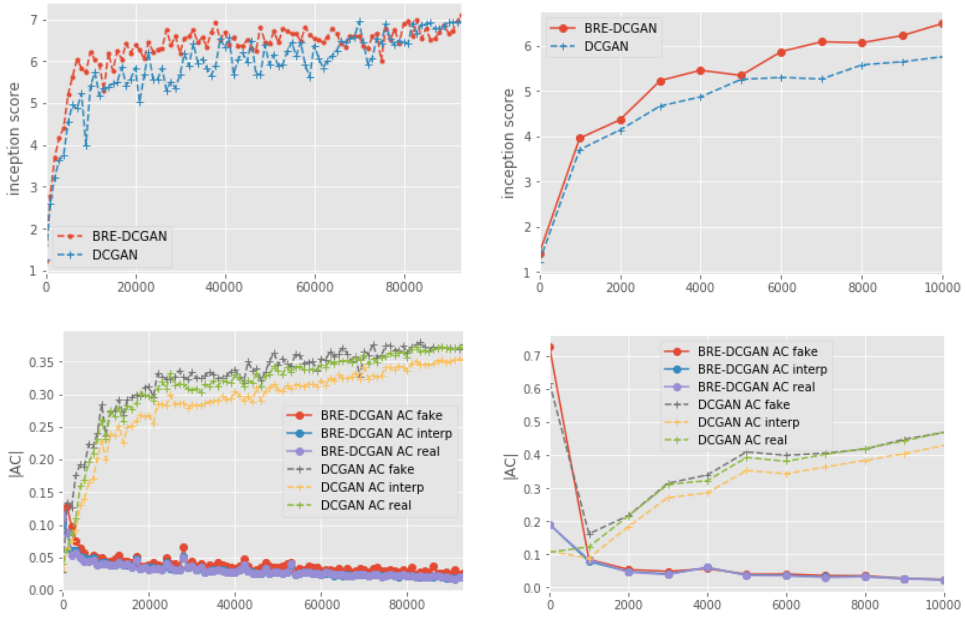

Figure 4: Inception scores and Regularizer values throughout training: (left column) $lr = 2e-4$ and 1 D update for 1 G update, lr for both D and G annealled to $1e-6$ over $90K$ G-updates; (right column) $lr = 2e-3$ and 3 D update for 1 G update, lr for both D and G annealled to $1e-6$ over $10K$ G-updates; (top row) inception scores during training; (bottom row) AC term of BRE on fake, interp and real data; Even though BRE is not applied on real, model still allocates enough capacity if BRE is applied on fake and interp. Both models are DCGAN for G and D with batchnorm but no other stabilization techniques.

We also show that the proposed BRE regularizer is not only compatible with semi-supervised learning using GANs, but also improves classification accuracy. Fig. 5 shows that BRE allows learning to discover a better solution during training, as shown by both training losses as well as test classification error rate. We used exactly the same code and hyperparameter from Salimans et al. (2016), and add R_{BRE} regularization on every other second layer, starting from the 2nd until 4 layers before the classification layer, with a regularizer weight of .01 and not decayed.

5 Conclusions

We proposed a novel regularizer in this paper to guide the discriminator in GANs to better allocate its model capacity. Based on the relation between the model capacity and the activation pattern of the network, we construct our regularizer to encourage a high joint entropy of the activation pattern of the discriminator D in a weak manner. Experimental results demonstrate the benefits of our new regularizer. There are still many unexplored avenues alone this line of research. For example, how can our new regularizer collaborate with other GANs training techniques to further improve the training GANs? We leave such further explorations for future works.

References

Sanjeev Arora, Rong Ge, Yingyu Liang, Tengyu Ma, and Yi Zhang. Generalization and equilibrium in generative adversarial nets (gans). *arXiv preprint arXiv:1703.00573*, 2017.

David Berthelot, Tom Schumm, and Luke Metz. Began: Boundary equilibrium generative adversarial networks. *arXiv preprint arXiv:1703.10717*, 2017.

Georges A Darbellay and Igor Vajda. Estimation of the information by an adaptive partitioning of the observation space. *IEEE Transactions on Information Theory*, 45(4):1315–1321, 1999.

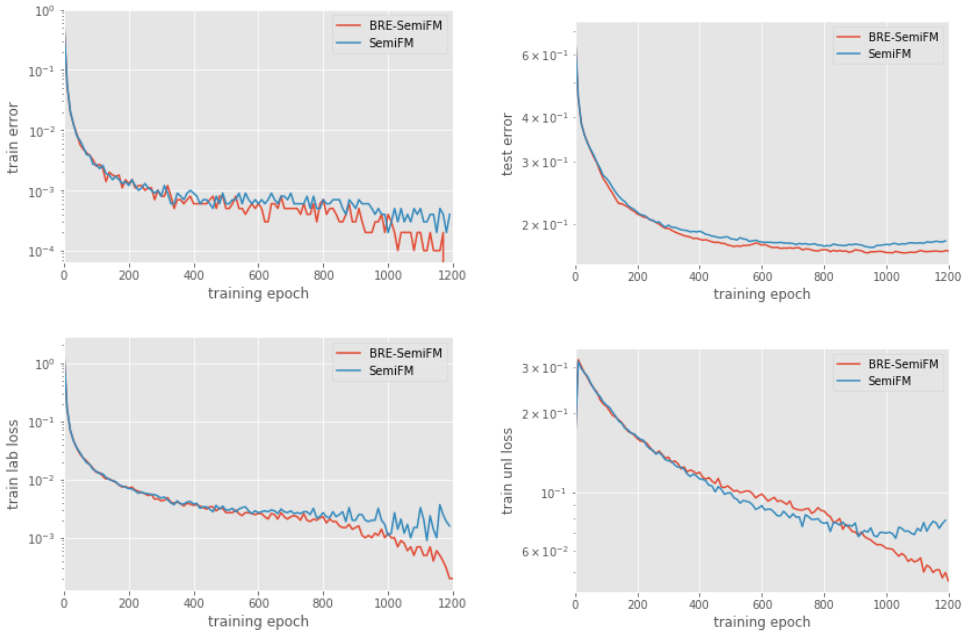

Figure 5: Improved semi-supervised learning on CIFAR-10: BRE regularizer placed on every other second layer, starting from the 2nd until 4 layers before the classification layer. Regularizer weight is .01 and not decayed.

Chelsea Finn, Paul Christiano, Pieter Abbeel, and Sergey Levine. A connection between generative adversarial networks, inverse reinforcement learning, and energy-based models. *arXiv preprint arXiv:1611.03852*, 2016.

Dmitry Gavinsky and Pavel Pudlák. On the joint entropy of d-wise-independent variables. *arXiv preprint arXiv:1503.08154*, 2015.

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*, pp. 2672–2680, 2014.

Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron Courville. Improved training of wasserstein gans. *arXiv preprint arXiv:1704.00028*, 2017.

Peter Henderson, Wei-Di Chang, Pierre-Luc Bacon, David Meger, Joelle Pineau, and Doina Precup. Optiongan: Learning joint reward-policy options using generative adversarial inverse reinforcement learning. *arXiv preprint arXiv:1709.06683*, 2017.

Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. *arXiv preprint arXiv:1611.07004*, 2016.

Alex Krizhevsky. Learning multiple layers of features from tiny images. 2009.

Jan Kybic. High-dimensional entropy estimation for finite accuracy data: R-nn entropy estimator. In *Information Processing in Medical Imaging*, pp. 569–580. Springer, 2007.

Jan Kybic and Ivan Vnučko. Approximate all nearest neighbor search for high dimensional entropy estimation for image registration. *Signal Processing*, 92(5):1302–1316, 2012.

Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 3730–3738, 2015.

Michael Mathieu, Camille Couprie, and Yann LeCun. Deep multi-scale video prediction beyond mean square error. *arXiv preprint arXiv:1511.05440*, 2015.

Lars Mescheder, Sebastian Nowozin, and Andreas Geiger. The numerics of gans. *arXiv preprint arXiv:1705.10461*, 2017.

Erik G Miller. A new class of entropy estimators for multi-dimensional densities. In *Acoustics, Speech, and Signal Processing, 2003. Proceedings.(ICASSP'03). 2003 IEEE International Conference on*, volume 3, pp. III–297. IEEE, 2003.

Guido F Montufar, Razvan Pascanu, Kyunghyun Cho, and Yoshua Bengio. On the number of linear regions of deep neural networks. In *Advances in neural information processing systems*, pp. 2924–2932, 2014.

Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.

Maithra Raghu, Ben Poole, Jon Kleinberg, Surya Ganguli, and Jascha Sohl-Dickstein. On the expressive power of deep neural networks. *arXiv preprint arXiv:1606.05336*, 2016.

Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. In *Advances in Neural Information Processing Systems*, pp. 2234–2242, 2016.

David W Scott. *Multivariate density estimation: theory, practice, and visualization*. John Wiley & Sons, 2015.

Ashish Shrivastava, Tomas Pfister, Oncel Tuzel, Josh Susskind, Wenda Wang, and Russ Webb. Learning from simulated and unsupervised images through adversarial training. *arXiv preprint arXiv:1612.07828*, 2016.

Junbo Zhao, Michael Mathieu, and Yann LeCun. Energy-based generative adversarial network. *arXiv preprint arXiv:1609.03126*, 2016.

2	d_width_factor:1	d_width_factor:2	d_width_factor:4	d_width_factor:8
mean	-0.012558	-0.095822	-0.144102	0.001150
min	-0.154278	-0.254397	-0.203341	-0.135540
25%	-0.075591	-0.208932	-0.183847	-0.071670
50%	0.003096	-0.163468	-0.164353	-0.007801
75%	0.058301	-0.016534	-0.114482	0.069495
max	0.113507	0.130399	-0.064611	0.146790
2	d_width_factor:1	d_width_factor:2	d_width_factor:4	d_width_factor:8
std	0.199080	0.163669	0.122830	0.243098
min	-0.205273	-0.176051	0.010553	-0.065005
25%	0.076265	0.045827	0.142776	0.159023
50%	0.257582	0.166419	0.284752	0.415546
75%	0.276390	0.244085	0.313710	0.476366
max	0.444160	0.358119	0.371657	0.655426

Table 1: Distribution of Inception Score improvement: initial transient phase vs convergence phase for different D size factor while holding G size constant. 1 D update for 1 G update with $lr = 2e-4$, initial transient phase before 5K iteration see Fig. 4(c). Inception scores are averaged across three repeated runs; With BRE regularizer, GAN training makes less agressive progress initially during the transient phase, but converges to better solution later.

A PROOF OF PROPOSITION 3.1

Proof. Let $M_k = U_k \tilde{U}_k$. Then

$$
\begin{aligned}
\mathbb{E}\left[\left(\sum_{k=1}^{d} U_k \tilde{U}_k\right)^2\right] &= \mathbb{E}\left[\left(\sum_{k=1}^{d} M_k\right)^2\right] \\
&= \sum_{k=1}^{d} \mathbb{E}\left[\, M_k^2 \,\right] + \sum_{k \neq j} \mathbb{E}\left[\, M_k M_j \,\right] \\
&= \sum_{k=1}^{d} \mathbb{E}\left[\, U_k^2 \tilde{U}_k^2 \,\right] + \sum_{k \neq t} \mathbb{E}\left[\, U_k \tilde{U}_k U_t \tilde{U}_t \,\right] \\
&\overset{(1)}{=} \sum_{k=1}^{d} \mathbb{E}\left[\, U_k^2 \,\right]^2 + \sum_{k \neq t} \mathbb{E}\left[\, U_k U_t \,\right]^2 \\
&\overset{(2)}{=} d + \sum_{k \neq t} \mathbb{E}\left[\, U_k U_t \,\right]^2 \\
&\overset{(3)}{=} d + \sum_{k \neq t} \mathrm{Cov}\left(U_k, U_t\right)^2,
\end{aligned}
$$

where Equation (1) is due the independence of U and $\tilde{U}$, Equation (2) is due to that $U_k^2 = 1$ with probability 1, and Equation (3) is because $\mathbb{E}\left[\, U_k \,\right] = 0$. □

B ADDITIONAL EXPERIMENTAL RESULTS

We show more 2D mixture of Gaussian results in Figure 6 and 7.

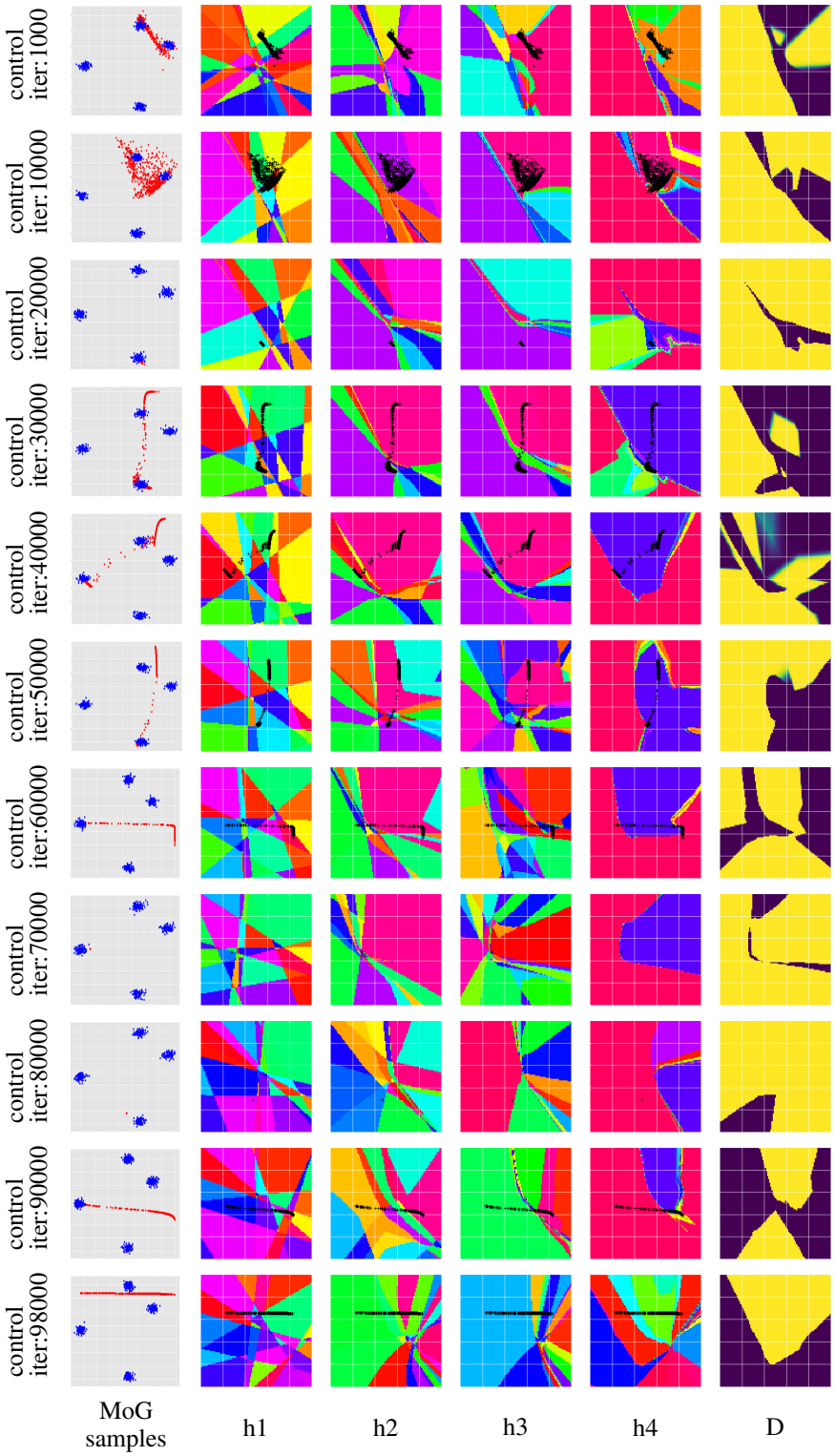

Figure 6: More Results on Fitting 2D Mixture of Gaussian on the control group. See Figure 2 for detailed description.

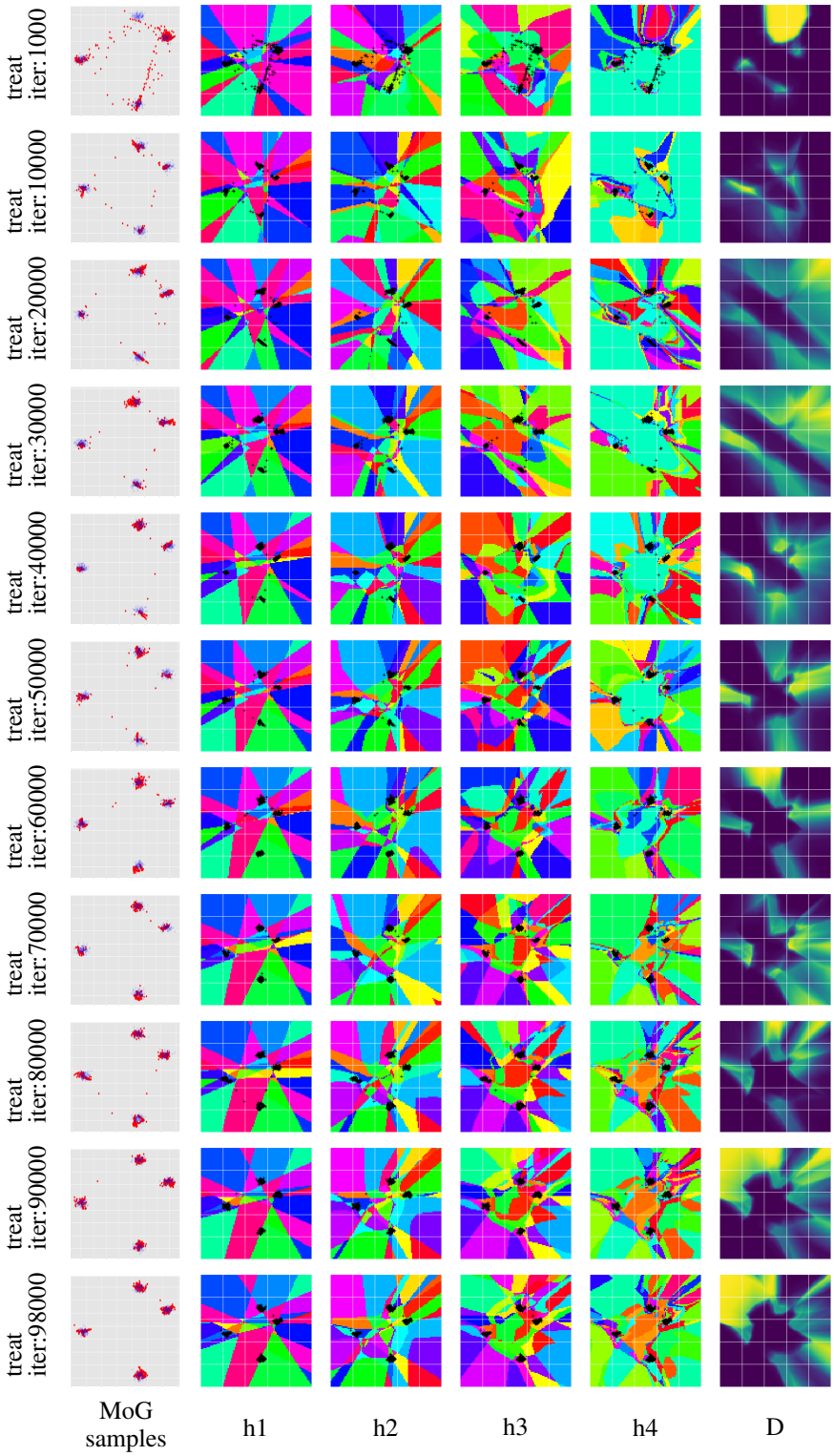

Figure 7: More Results on Fitting 2D Mixture of Gaussian on the treat group. See Figure 2 for detailed description.