Mixture-GAN
Final Report

Jaspal Singh
Department of Computer Science
Yale University
New Haven, CT 06511
jaspal.singh@yale.edu

Alexander Tong
Department of Computer Science
Yale University
New Haven, CT 06511
alexander.tong@yale.edu

1 Summary of Current Progress

We originally set out to answer the question of how GANs perform on biased data. We defined biased data in terms of the weighted mixture of two gaussians. In effect, we looked at the Li et al. [3] with an optimal discriminator, because it is a simple model that captures the convergence effects of GANs. Specifically, we were interested in how the relative mixture of the two gaussians affects convergence rates under this model, and hopefully by extension, to more realistic models. As a reminder, we were interested in the following generator:

$$\mathcal{G} = \left\{ \alpha\mathcal{N}(\mu_1, 1) + (1-\alpha)\mathcal{N}(\mu_2, 1) \mid \mu_1, \mu_2 \in \mathbb{R} \text{ and } \alpha \in [0,1] \right\} \tag{1}$$

We consider only the **Known** α case in this paper. The known case is assuming apriori knowledge of the value of α. In a more realistic setting this is a very weak form of supervision, as it is often easier to know approximately the overall statistics of a dataset than labels for each datapoint. We study the dynamics under an optimal discriminator and a first order discriminator (potentially with unrolling) for this setting.

A summary of the progress made in the project:

1. Implementation of code reproducing the results of Li et al. (2018) and extending the code for the known α setting with an altered learning rate version.
2. Evaluation of the convergence probability for varying dynamics of the Li et al. model.
3. Study of the convergence time and probability as a function of the weight parameter α in the Li et al. model and a network model.
4. Explored the convergence of a neural network based GAN a 2d gaussian mixture model showing one particular cause of mode collapse in this instance.

Code associated with this work can be found at: `https://github.com/atong01/mixture-gan/`

2 Model Description

Loss function While there are many loss functions that one can choose from, in this project we will mainly focus on the total variation distance defined as: For distribution P, Q,

$$d_{TV}(P, Q) = \max_A P(A) - Q(A) \tag{2}$$

where the maximum is taken over all measurable sets.

This exact loss function was also considered in [3] since it simplifies for the Gaussian mixture model to the following:

$$d_{TV}(G_{\mu,\alpha}, G_{\mu^*,\alpha^*}) = \max_{E=I_1 \cup I_2} G_{\mu,\alpha}(E) - G_{\mu^*,\alpha^*}(E) \tag{3}$$

where the maximum is taken over the union of two disjoint intervals $I_1, I_2 \subset \mathbb{R}$. While this simplification is not explicitly described in [3] it trivially follows from their Theorem A.2.

Discriminators The more simplified formulation of the loss function given in Equation 3 allows for the following very natural definition of the discriminator:

$$\mathbb{D} = \{\mathbb{I}_{l_1,r_1} + \mathbb{I}_{l_2,r_2} \mid l_1, r_1, l_2, r_2 \in \mathbb{R} \text{ and } l_1 \leq r_1 \leq l_2 \leq r_2\} \tag{4}$$

where $\mathbb{I}_{l,r}$ is the indicator function that outputs 1 on input in range $[l, r]$ and outputs 0 otherwise. A discriminator from this set with parameters l_1, r_1, l_2, r_2 is represented by $D_{l,r}$, where $l = (l_1, l_2)$ and $r = (r_1, r_2)$.

Hence, the loss function can be re-written as:

$$L(\mu, l, r) = \left(\int_I G_{\mu^*,\alpha}(x) - G_{\widehat{\mu},\alpha}(x) dx \right) \text{ where } I = [l_1, r_1] \cup [l_2, r_2] \tag{5}$$

$$\text{and } G_{\mu,\alpha}(x) = \frac{1}{\sqrt{2\pi}} \left(\alpha e^{-(x-\mu_1)^2/2} + (1-\alpha) e^{-(x-\mu_2)^2/2} \right) \tag{6}$$

Problem statement Consider the data distribution P equal to $\alpha\mathcal{N}(\mu_1^*, 1) + (1-\alpha)\mathcal{N}(\mu_2^*, 1)$, for some $\alpha \in [0, 1]$ and $\mu_1^*, \mu_2^* \in \mathbb{R}$. The objective is to solve the following min-max optimization problem:

$$\begin{aligned} \hat{\mu} &= \underset{\mu}{\operatorname{argmin}} \max_{l,r} L(\mu, l, r) \\ &= \underset{\mu}{\operatorname{argmin}} \max_{l,r} \mathbb{E}_{x\sim P}[D_{l,r}(x)] + \mathbb{E}_{x\sim G_{\mu^*}}[1 - D_{l,r}(x)] \end{aligned} \tag{7}$$

To solve the above described problem we will aim to analyze the performance of three classes of algorithm based on first order dynamics, optimal discriminator dynamics, and optimal discriminator dynamics with altered learning rates.

First order dynamics In this approach we solve the min-max optimization problem defined in Equation 7 by iteratively applying gradient descent. We minimize the function along the parameters l, r followed by maximizing along the parameters μ. Formally, the first order dynamics are specified by the following equations:

$$\begin{aligned} &\text{Given starting points } \mu^{(0)}, l^{(0)}, r^{(0)} \text{ and step sizes } \eta_g, \eta_d : \\ &\mu^{(t+1)} = \mu^{(t)} - \eta_g \nabla_\mu L(\mu^{(t)}, l^{(t)}, r^{(t)}) \\ &l^{(t+1)} = l^{(t)} - \eta_d \nabla_l L(\mu^{(t)}, l^{(t)}, r^{(t)}) \\ &r^{(t+1)} = r^{(t)} - \eta_d \nabla_r L(\mu^{(t)}, l^{(t)}, r^{(t)}) \end{aligned}$$

Optimal discriminator dynamics In this approach we perform gradient descent on the function over the parameters of the generator (for solving the "min" component of the problem) while assuming the optimal discriminator at each step (i.e. solving the "max" component of the problem optimally). More formally the optimal discriminator dynamics can be given by:

$$\begin{aligned} &\text{Given starting point } \mu^{(0)} \text{ and step size } \eta_g : \\ &l^{(t)}, r^{(t)} = \underset{l,r}{\operatorname{argmax}} L(\mu^{(t)}, l, r) \\ &\mu^{(t+1)} = \mu^{(t)} - \eta_g \nabla_\mu L(\mu^{(t)}, l^{(t)}, r^{(t)}) \end{aligned}$$

Let $\alpha_1 = \alpha$ and $\alpha_2 = 1 - \alpha$. Then the gradients of the loss function are:

$$\frac{\partial}{\partial \widehat{\mu}_j} L = \alpha_j \frac{1}{\sqrt{2\pi}} \sum_{i=1,2} e^{-(r_i - \widehat{\mu}_j)^2/2} - e^{-(l_i - \widehat{\mu}_j)^2/2}$$

$$\frac{\partial}{\partial l_i} L = -\alpha_j \frac{1}{\sqrt{2\pi}} \sum_{j=1,2} e^{-(l_i - \mu_j^*)^2/2} - e^{-(l_i - \widehat{\mu}_j)^2/2}$$

$$\frac{\partial}{\partial r_i} L = \alpha_j \frac{1}{\sqrt{2\pi}} \sum_{j=1,2} e^{-(r_i - \mu_j^*)^2/2} - e^{-(r_i - \widehat{\mu}_j)^2/2}$$

Optimal discriminator dynamics with altered learning rates As depicted by the gradients above, all gradients are multiplied by the weight parameter α_1 or α_2. If the weights are unbalanced it leads to different rates of improvement in μ_1 or μ_2. So as to reduce this influence of the weight parameter on the gradients, we considered a modified optimal discriminator dynamics where the learning rate $\eta_i = \eta/\alpha_i$ for $i = 1, 2$. Hence, the formal definition is as follows:

$$\begin{aligned}
&\text{Given starting point } \mu^{(0)} \text{ and step size } \eta : \\
&l^{(t)}, r^{(t)} = \operatorname*{argmax}_{l,r} L(\mu^{(t)}, l, r) \\
&\mu_1^{(t+1)} = \mu_1^{(t)} - \eta_1 \nabla_{\mu_1} L(\mu^{(t)}, l^{(t)}, r^{(t)}) \\
&\mu_2^{(t+1)} = \mu_2^{(t)} - \eta_2 \nabla_{\mu_2} L(\mu^{(t)}, l^{(t)}, r^{(t)})
\end{aligned}$$

3 Experimental Results

We experimented using a number of different models to gain a better understanding of GANs on a biased dataset. We approached the problem from two angles, using a realistic model with generated data, and using a less realistic but more interpretable model as introduced in Li et al. to study GAN behavior in general. We hoped to understand the convergence behavior of a GAN using Li et al.'s model by simply adapting it to an uneven mixture 7, as the authors previously had success in showing a mode dropping free case (optimal discriminator).

While we probably should have done so in the opposite order, we present our results somewhat chronologically. We first implemented Li et al.'s model, then adapted it to a varied α case. We explored this case for a while only realizing after quite a while that there was a bug in our gradient calculation code and we were analyzing a slightly different case than we though.

We first analyze three cases with the Li et al. model, then show results on a more realistic network based model.

3.1 Li et al. Gaussian Mixture

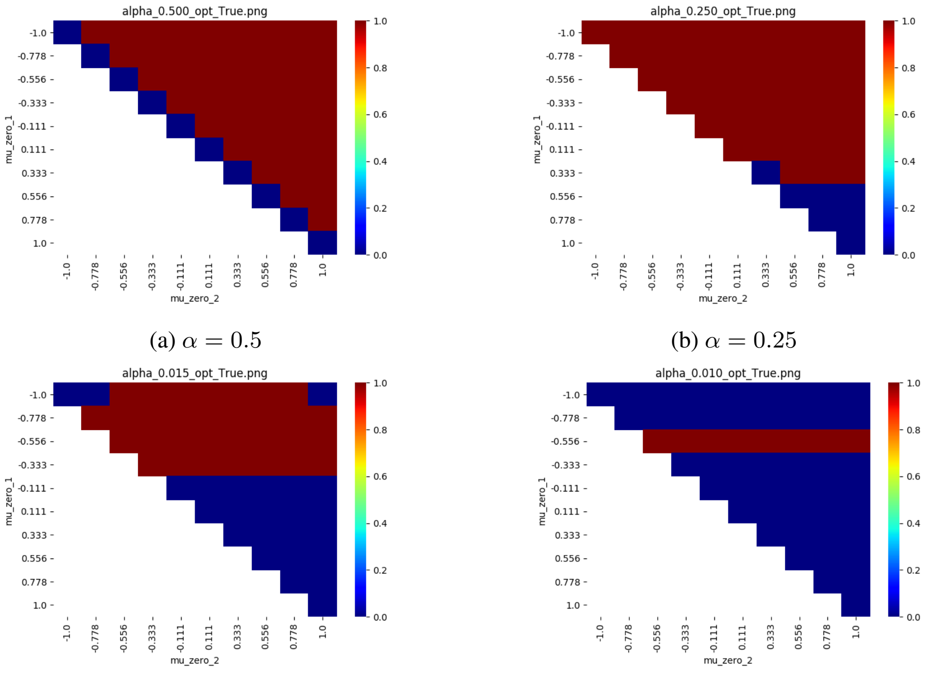
Here we study the effects of the generator as described in equation 1, but for three different learning procedures, (1) Optimal discriminator, (2) Optimal discriminator with altered learning rates, (3) First order discriminator with altered learning rates.

In the optimal discriminator case we found a case where the order of the modes can initially switch to the wrong order, then get stuck in a local optima for the generator and never converge to the correct values. We then examine the case where we alter the learning rate for μ_1 vs μ_2 based on the known value of α. We examine the case where we multiply the learning rate for μ_j by α_j^{-1}. We come to this case by inspection of the gradient of μ. Namely by inspection of the gradient in equation 21, we can see by multiplying by this value for each μ we are effectively normalize out the α parameter in the function, and we observe much more efficient gradient descent steps, and notably it seems to always converge. with this normalization.

Fixed α Under Optimal Discriminator The most straightforward adaptation of the Li et al. model to the biased case leaves all the training parameters the same, and simply inserts a known mixture parameter α into the generator and discriminator. We note by inspection of the gradients in equation 21, that the gradient of μ_j is a function of $\alpha_j, l, r,$ and μ_j. Specifically, it is always multiplied by α_j. Thus in the extremely biased case where α_j is close to zero, the gradient for μ_j is very small. This is our first hint that this model may not converge in a reasonable number of steps always.

We found a case where this model never converges, namely if the modes are in the "wrong order" then there is a local minima for the generator, even under optimal discriminator dynamics that causes it to settle into a local minima with the modes in the incorrect order, and subsequently never converge to the ground truth values of μ. This can be seen in Figure 2. In this figure, we essentially observe that μ_2 settles to the value 0.22, which is between the 2 μ^* values, whereas μ_1 is very slow to converge.

Looking at the full heatmap we see that convergence probability is dominated by the location of the μ with the smaller value of α, namely as the distribution becomes more biased, the range of acceptable initial values of μ_{small} decreases. Figure 1a shows that if $\alpha = 0.5$ then the model suffers from mode collapse, namely if the two values $\widehat{\mu_1} = \widehat{\mu_2}$ then the optimal discriminator dynamics do not converge. If $\alpha \neq 0.5$ then this does not occur and even if we initialize $\widehat{\mu_1} = \widehat{\mu_2}$ we may still get convergence. For uneven dynamics we do not see convergence if both $\widehat{\mu}$ values are greater than μ_2.



(a) $\alpha = 0.5$

(b) $\alpha = 0.25$

(c) $\alpha = 0.015$

(d) $\alpha = 0.01$

Figure 1: Shows the convergence for different initial $\widehat{\mu}$ and different α levels.

Fixed α Under Optimal Discriminator Dynamics with Altered Learning Rates We focus our remaining attention on the case that converges (experimentally) for all initial values of μ as long as the $\widehat{\mu}$ are in the correct order, i.e. guess of the μ with the smaller value of α is on the same side of the μ with the larger value of α as in the true generated distribution. In Figure 4 we can see that even with a large skew in α dynamics under the optimal discriminator always converge to the correct value. We tested α values 10 values of α using a geometric distribution in the range $(0.01, 0.5)$, and $\widehat{\mu}$ initializations linearly spaced in the range of $[-1, 1]$. Over these 450 tests we found that when values of μ were initializied to the same value (i.e. mode collapsed) then the model did not converge. However, when we initialized $\widehat{\mu}$ to any other set in the correct order, we observed convergence. This case is then most similar (at least experimentally to that in the optimal discriminator analysis of Li et al..

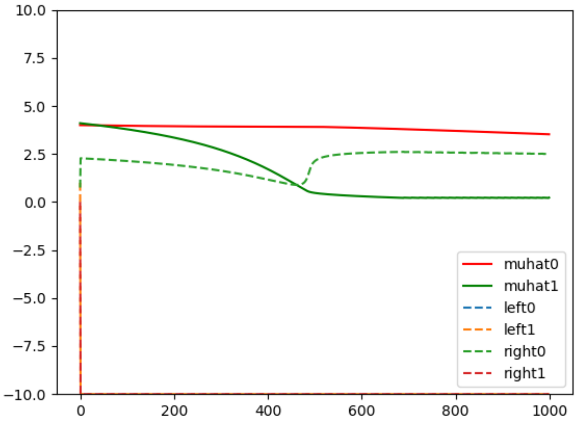


Figure 2: Optimal discriminator dynamics for parameters $\alpha_1 = 0.1, \alpha_2 = 0.9, \mu^* = (0, 0.5), \mu^{(0)} = (4, 4.1)$. The μ orders are reversed and μ_2 converges to 0.22 while μ_1 is very slow to converge

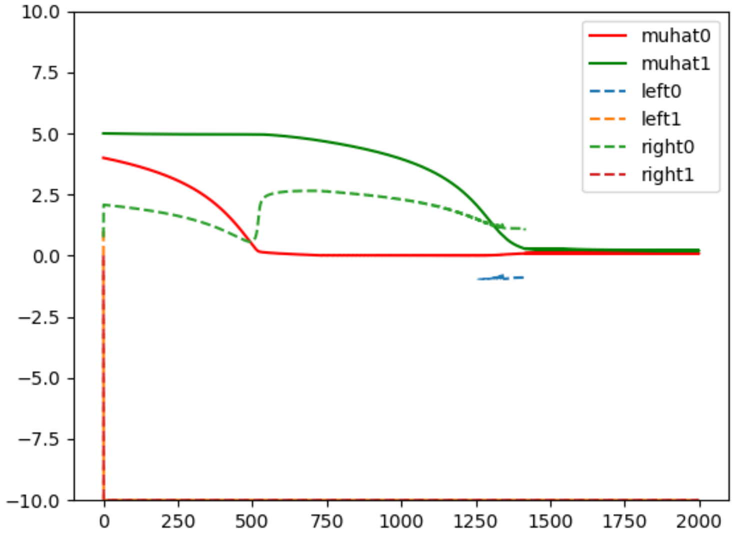
Figure 3 shows how the μ parameter qualitatively converges to the optimal value μ^* for different α parameters. For the case $\alpha = (0.1, 0.9)$ (Figure 3a), μ_1 converges first at around 500 iterations, and that is when μ_2 starts improving at a significant rate as well until it converges. This convergence behaviour has been observed over a large range of initial μ and μ^* values as well, though we don't know the exact reason for the same. For the case $\alpha = (0.1, 0.9)$ (Figure 3b), the behaviour is different is a way that μ_1 overshoots from the optimal value and let μ_2 converge before itself. What we can conclude experimentally from both these cases is that the μ value corresponding to the higher weight parameter always converges first.

Rate of convergence with Altered Learning Rates Figure 5 shows the the number of iterations it takes for the means $\widehat{\mu}$ to get within some δ distance to the optimal μ^*. Each iteration consists of first order updates over $\widehat{\mu}$ and optimal updates over discriminator parameters l, r. As can be seen from the plot, though optimal discriminator with altered learning rates always converges, its speed of convergence seems to be inversely proportional to $\alpha(1-\alpha)$. We can further question here on how to make the convergence rate independent of α by setting the η parameter accordingly. As in Figure 3 We can see that the μ with the larger α value converges first, then the μ with the smaller α begins to converge. This might explain the slower convergence for more biased values for α.

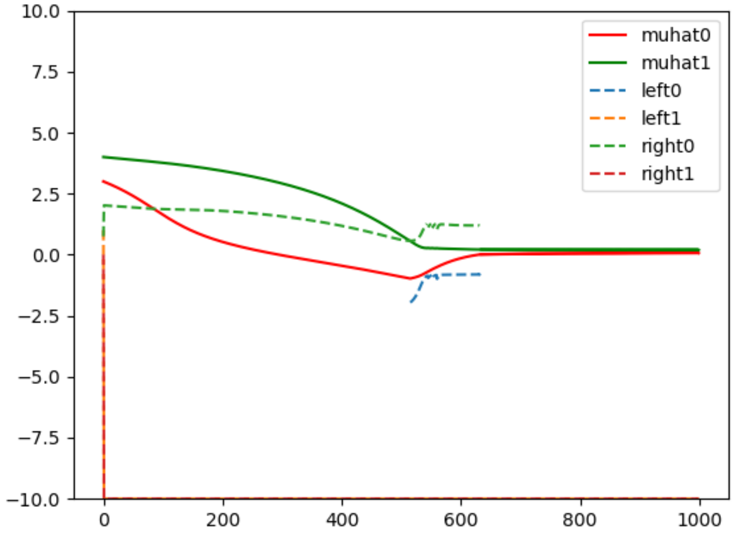
Fixed α Under First Order Dynamics Fixed α under first order dynamics can be shown for various levels of α in Figure 6. For $\alpha = 0.5$ we roughly recover the results of Li et al. Figure 2a in terms of convergence probabilities. We take all parameters except for the distribution of initial discriminator intervals which is referred to as "at random". We suspect this leads to small differences in convergence probabilities.

From these plots suspect the following. Slightly uneven weighting gives higher success probability (See Figure 6b) than perfectly even weighting. At extremely low values of α we only succeed on initializations of α where the smaller mean is initialized to below its optimum i.e. $\widehat{\mu_1} < \mu_1^*$ when $\alpha_1 << \alpha_2$. This was unexpected and currently not fully explained.

Note on the Initialization Order of μ Since we are considering the one dimensional case, when we look at the known α condition we can consider knowing the order of α, i.e. whether the gaussian with larger α^* has larger or smaller μ^* value. In our experiments we fixed the case that our initialization has the correct ordering, i.e. we do not have to flip the order of the gaussians (as we are not learning α). This simplifies the problem but does not generalize to higher dimensions or number of gaussians. We tried the optimal discriminator with the incorrect initialization order and this failed 100% of the time.



(a) $\alpha = (0.1, 0.9)$



(b) $\alpha = (0.9, 0.1)$

Figure 3: For $\mu^* = (0.1, 0.2)$ the plots show the convergence of μ for diffeent weight parameters α

3.2 Network Model

While the Li et al. model is useful in that it is interpretable, we wanted to examine the behavior on a more realistic GAN. Our true data model is two 2d gaussian distributions centered at -2,0 and 2,0 with varying standard deviations and α values. The true generated distribution is summarized in the following equation:

$$p(x) \sim \alpha\mathcal{N}([-2,0]\Sigma) + (1-\alpha)\mathcal{N}([2,0],\Sigma) \mid \Sigma \in \text{diag}(\mathbb{R}^2) \text{ and } \alpha \in [0,1] \quad (8)$$

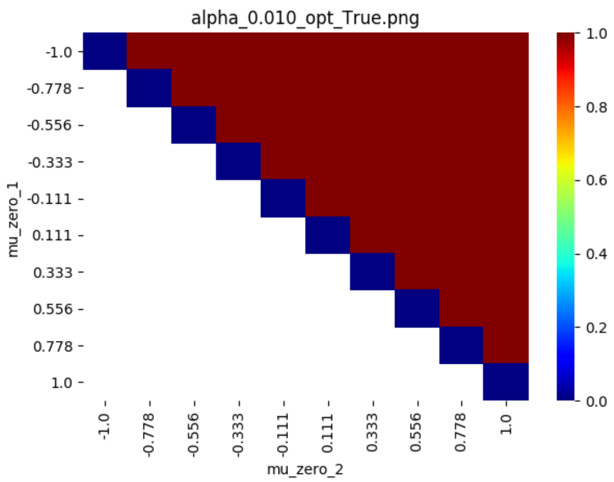


Figure 4: Convergence probabilities over initial values of $\widehat{\mu}$ with $\alpha = 0.01$

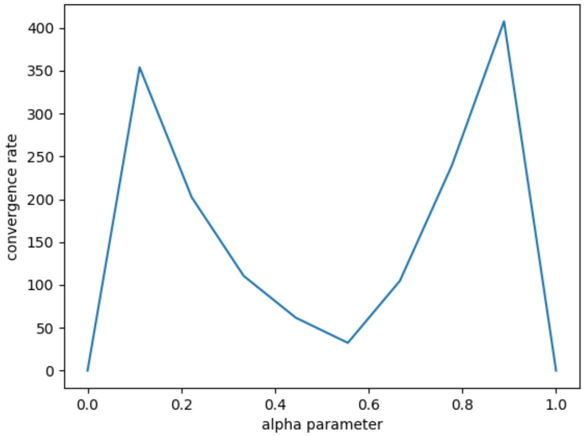
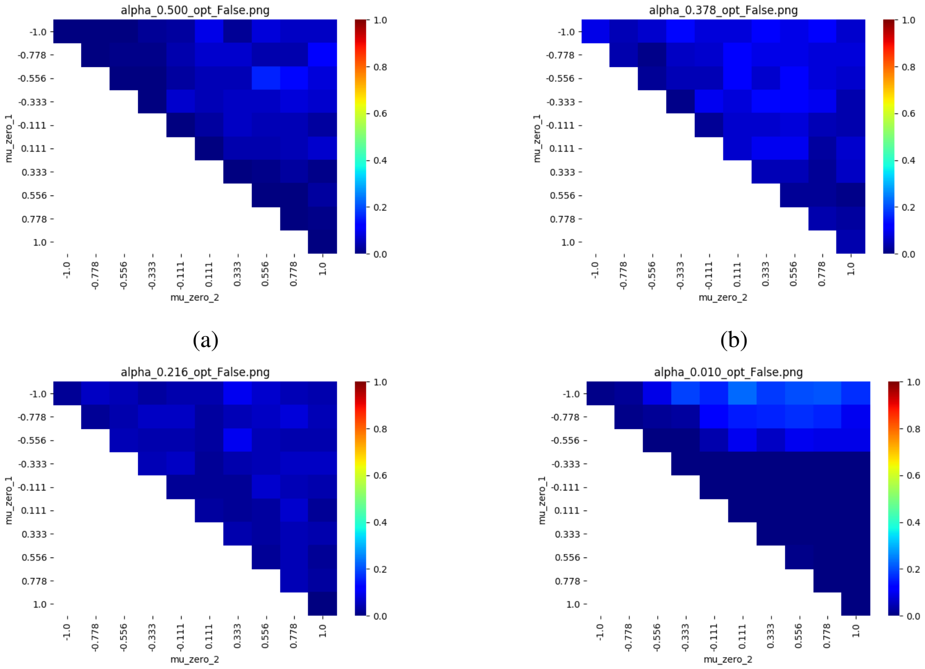


Figure 5: Plot of average rate of convergence for optimal discriminator dynamics vs known α with initialization $\mu^* = (-0.5, 0.5), \alpha \in Linspace(0, 1, 10)$ and the tolerance parameter $\delta = 0.01$

We note that this is equivalent to the case in equation 1, lifted into two dimensions. We alter Σ instead of μ in this case so that our (1) statistical distance measure is comparable and (2) initialization of the model weights is not a large source of variation in our comparison. Figure 7 illustrates this data distribution.

Convergence Rates α, σ Next we examine the effect of α and σ on the convergence rate of the network model. We use the Wasserstein distance between 10,000 generated samples and 10,000 true samples as a proxy for the quality of the generator. From Figure 8a we can see that the more balanced α is faster the convergence rate, and the better the overall quality at 10,000 iterations. With gaussians that are further apart. We note that at values of $\alpha < 0.05$ we often see mode collapse when looking at generated samples (this is explored further in Figure 9). Looking over varying values of σ, the smaller σ whos distributions should be easier to learn, the closer the biased α values are as compared to the unbiased. This implies that as the distribution is easier to learn, the value of α may have a smaller effect on learnability and convergence.

Mode Collapse for Biased α In extremely biased distributions we see mode collapse in the generator. We define this as only well approximating samples from one of the two modes at 10,000



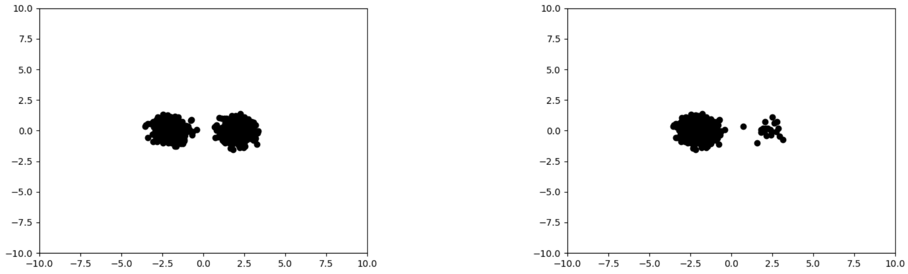
(a)

(b)

(c)

(d)

Figure 6: Shows the convergence probabilities for various settings of α under first order dynamics. Note that the value of α refers to the weight of the μ_1^*, so plots (b-d) are asymmetric. We average each experiment over 100 choices of initial discriminator intervals sampled from $U(-2,2)$ then sorted. Note that these values of α are selected from $GeomSpace(0.01, 0.5, 10)$.



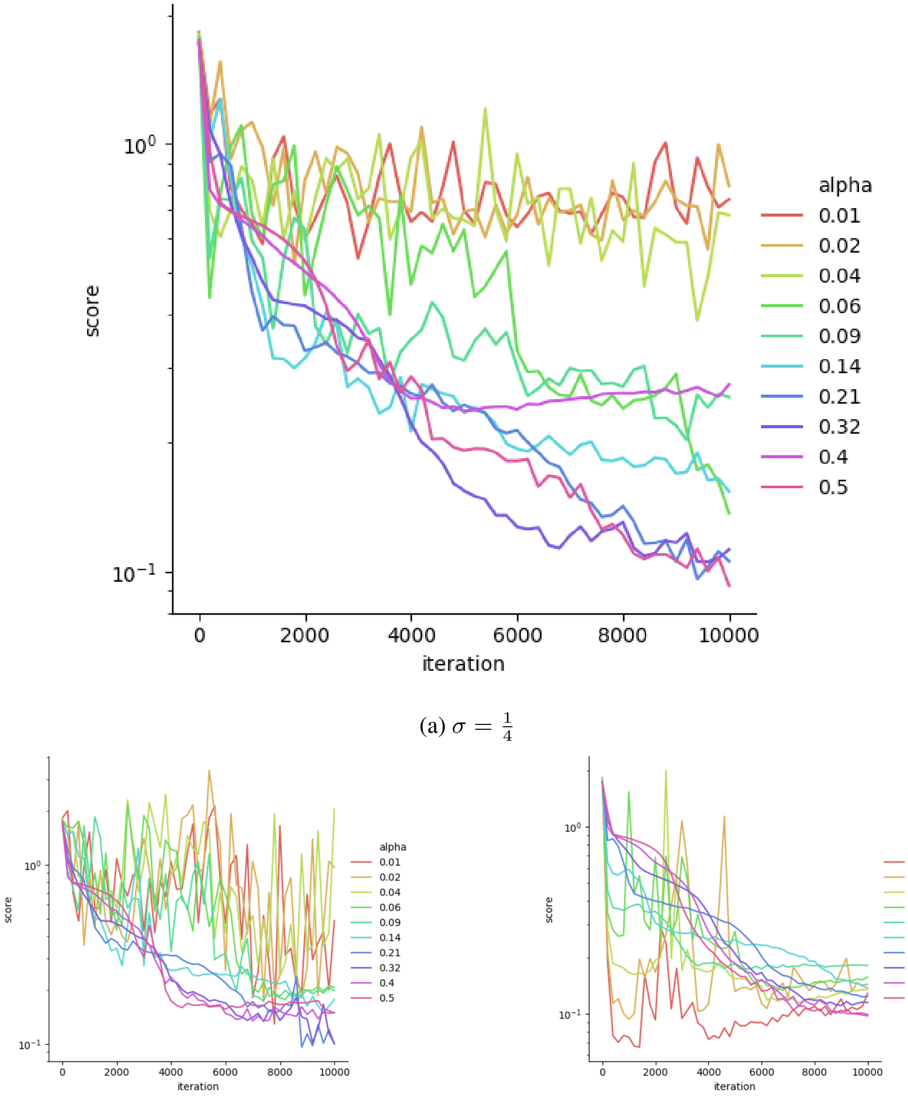
(a)

(b)

Figure 7: Shows 1000 points generated from 2d gaussian mixture with (a) $\alpha = 0.5$ (b) $\alpha = 0.98$ and $\Sigma = [[0.25, 0], [0, 0.25]]$

iterations. In Figure 9 we investigate this effect further for one model with $\alpha = (0.98, 0.02)$. In Figure 9a we can see the discriminator values, which suggest why the generator is stuck in a local minimum at 10,000 iterations. In the discriminator function, low values equate to low probability of data in that space. We can see a vertical line of low probability region on the left side of the generated distribution around $x = 0$. Since the generator must learn a continuous manifold, before it can produce an point near $+2$ it must produce points near zero. Since the discriminator function discourages this, then the generator will never move some of its probability mass in the correct direction.

We note that this is quite a different reason for mode collapse than in the Li et al. model. Here we have a generator that is constrained to make small local updates to its probability mass function in the output space. **Even with the optimal discriminator in this case, we would not make generator**



(a) $\sigma = \frac{1}{4}$

(b) $\sigma = \frac{1}{8}$

(c) $\sigma = \frac{1}{32}$

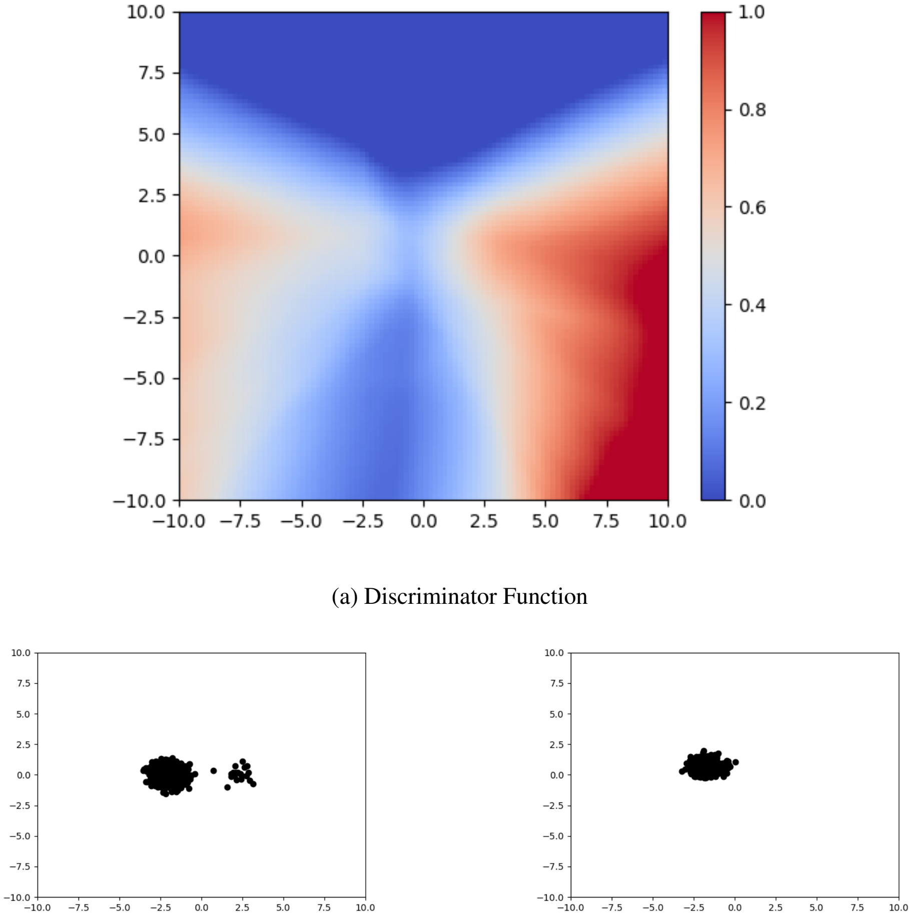
Figure 8: Shows the training convergence of our GAN run for 10,000 iterations using the 1-Wasserstein distance betweeen 10,000 generated points from the generator and 10,000 true data points for three different values of σ. Here lower score implies better convergence. $\Sigma = [[\sigma, 0], [0, \sigma]]$ where (a) $\sigma = \frac{1}{4}$ (b) $\sigma = \frac{1}{8}$, (c) $\sigma = \frac{1}{32}$.

updates in the correct direction. This suggests that the Li et al. model may not capture the salient details in this particular more realistic example, as it never mode collapses in the optimal discriminator case for balanced α. One of the causes of mode collapse is the continuous nature of generators, and the local minimum in the generator acting as a barrier to a better global minimum for the generator[1].

4 Conclusion

We studied the effect of biased datasets in GAN training for both a simple model as in Li et al. [3], and for a more realistic neural network model. In both models we found that the convergence probability was negatively impacted by the bias of the data, but for different reasons. In Li et al.

[1] The optimal discriminator for a 1-wasserstein objective function would fix the problem in this specific case.



(a) Discriminator Function

(b) True distribution

(c) Generator distribution

Figure 9: Shows an example of mode collapse with $\alpha = 0.98$. (a) Shows the value of the discriminator over the real plane after 10000 iterations. (b) Shows the true distribution. (c) Shows the 1000 samples from the generator distribution after 10000 training iterations.

model, because the generator is supported over the entire measure, with an optimal discriminator based on TV distance the generator always converges. In the network model, we found that mode collapse happens at a certain threshold of $\alpha = 0.05$, but above that threshold convergence time is not well correlated with α. In mode collapse situations, using the JS-divergence objective function causes local minima even in the optimal discriminator function, which causes the generator (with finite support) to never converge to the global minima, the true distribution.

We observed many other interesting behavior of these types of GANs, but unfortunately ran out of time in building the theory around these observations. In future work we would like to guarantee convergence under altered learning rate and optimal discriminator, and explore theoretically why and when fixed α under optimal discriminator with standard learning rate fails.

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. *arXiv:1606.03657 [cs, stat]*, June 2016.

[2] Guang-Yuan Hao, Hong-Xing Yu, and Wei-Shi Zheng. MIXGAN: Learning Concepts from Different Domains for Mixture Generation. *arXiv:1807.01659 [cs, stat]*, July 2018.

[3] Jerry Li, Aleksander Madry, John Peebles, and Ludwig Schmidt. On the Limitations of First-Order Approximation in GAN Dynamics. In *Proceedings of the 35th International Conference on Machine Learning*, Stockholm, Sweeden, 2018.

A Definitions and Notations

- $\mu = (\mu_1, \mu_2) \in \mathbb{R}^2$ and $\alpha \in [0, 1]$
- $G_\mu(x) = \frac{1}{\sqrt{2\pi}} e^{-(x-\mu)^2/2}$
- $G'_{\mu,\alpha}(x) = \alpha G_\mu(x)$
- $L_{\mu^*}(\widehat{\mu}, l, r) = \left(\int_I G_{\mu^*}(x) - G_{\widehat{\mu}}(x) dx \right)$ where $I = [l_1, r_1] \cup [l_2, r_2]$
- $L'_{\mu^*,\alpha}(\widehat{\mu}, l, r) = \left(\int_I G'_{\mu^*,\alpha}(x) - G'_{\widehat{\mu},\alpha}(x) dx \right)$ where $I = [l_1, r_1] \cup [l_2, r_2]$
- $f_{\mu^*}(\mu) = \max_{l,r} L_{\mu^*}(\mu, l, r)$
- $f'_{\mu^*,\alpha}(\mu) = \max_{l,r} L'_{\mu^*,\alpha}(\mu, l, r)$
- Rect(δ) = $\{\mu : |\mu - \mu_i^*| < \delta$ for some i, j $\}$
- $\nabla_{\mu_j} f_{\mu^*}(\mu) = \frac{1}{\sqrt{2\pi}} \sum_{i=1,2} (e^{-(r_i - \widehat{\mu}_j)^2/2} - e^{-(l_i - \widehat{\mu}_j)^2/2})$
- $\nabla_{\mu_j} f'_{\mu^*,\alpha}(\mu) = \alpha_j \nabla_{\mu_j} f_{\mu^*}(\mu)$
- $\nabla_\mu f_{\mu^*}(\mu) = \begin{bmatrix} \nabla_{\mu_1} f_{\mu^*}(\mu) \\ \nabla_{\mu_2} f_{\mu^*}(\mu) \end{bmatrix}$
  we can define $\nabla_\mu f_{\mu^*,\alpha}(\mu)$ similarly.
- $B(C)$ represents the box of side length C around origin.
- $Sep(\gamma) = \{(v_1, v_2) \in \mathbb{R}^2 : |v_1 - v_2| > \gamma\}$
- Let $m = \min(\alpha, 1 - \alpha)$ and $M = \max(\alpha, 1 - \alpha)$
- $||\ ||$ represents the $L2$ norm.

B Learning α by First-Order Methods

In our original project proposal we suggested studying the case where we do not know the mixture parameter α parameter in the Li et al. model. We experimented a bit with this model, but ultimately abandoned it as it strayed further from a realistic model, and the fixed α case seemed to be difficult enough to analyze. In our more realistic case using a trained neural network in the `keras` framework we analyze the case of unknown α, as our GAN architecture was not able to easily capture a known prior on α and would need modifications like those in InfoGAN [1] or in Mix-GAN [2].

In Figure 6 we show what happens when we naively learn α using gradient descent with the same parameters as we used to learn μ and the first order discriminator. We implemented this case using the same learning rate for α as $\mu, l,$ and r. In this figure we can see that the value of α oscillates wildly and outside of the range of $(0, 1)$, its meaningful range. There are ways to constrain the value of α in this range using some function like learning on the inverse logistic function or utilizing some boundary condition on α. However, we felt that going in this direction was not useful for our project as we hoped from some theoretical analysis from the Li et al. direction.

C Failed Theoretical Analysis

We failed to produce meaningful theoretical results from our models. We spent a majority of our time analyzing the simple case with unadjusted η values. However, we were unable to prove certain

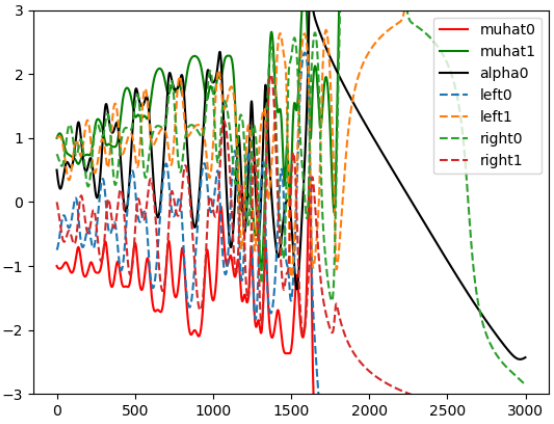


Figure 10: Diverging α_0 for first order learned α with initialization $\widehat{\alpha} = (0.5, 0.5)$, $\alpha^* = (0.5, 0.5)$, $\widehat{\mu}_0 = (-1, 1)$, $\mu^* = (-0.5, 0.5)$

things, in particular, mode collapse. We thought we could prove this because we had a bug in our code, namely our code was actually performing the adjusted version of η while we thought it was not. This led us to believe that it was possible to prove the convergence in the standard fixed α case, which by our recent experiments, is impossible, as this model does not always converge (see figure 1).

We now believe that the correct theorem to prove is for the adjusted η case, as it seems that this always converges (with some explainable caveats). However, we ran out of time to do so. This requires extensive case analysis as in the supplement of Li et al. but for approximately double the number of cases.

We realized that our method for extending existing results does not apply as we expected because the discriminator intervals are a function of α. Nevertheless, we believe that Lemmas concerning mode collapse and diverging behavior (Li et al. 4.5, 4.6) are easy to extend, (4.1, 4.2) would apply in entirety, and the tricky bit would be the case analysis in Li et al Lemmas 4.3, 4.4.

D Gradient Derivations

For convenience, define $F(\alpha^*, \widehat{\alpha}, \mu^*, \widehat{\mu}, x)$ as follows:

$$F(\alpha^*, \widehat{\alpha}, \mu^*, \widehat{\mu}, x) = \frac{1}{\sqrt{2\pi}} \sum_{j=1,2} \alpha_j^* e^{-(x-\mu_j^*)^2/2} - \widehat{\alpha}_j e^{-(x-\widehat{\mu}_j)^2/2} \tag{9}$$

Looking at the problem more precisely, we consider the loss function $L(\mu, l, r, \alpha)$. Note that we can consider the case where we know α, or we would like to learn α.

$$
\begin{aligned}
L(\mu, l, r, \alpha) &= \mathbb{E}_{x \sim G_{\mu^*}}[D(x)] + \mathbb{E}_{x \sim G_{\mu}}[1 - D(x)] && (10)\\
&= \left(\int_I G_{\mu^*}(x) - G_{\widehat{\mu}}(x) dx \right) + 1, && (11)
\end{aligned}
$$

Where $I = [l_1, r_1] \cup [l_2, r_2]$. We then have

$$
\begin{aligned}
&= \left(\sum_{i=1,2} \int_{l_i}^{r_i} G_{\mu^*}(x) - G_{\widehat{\mu}}(x) dx \right) + 1, && (12)\\
&= \frac{1}{\sqrt{2\pi}} \sum_{i=1,2} \sum_{j=1,2} \int_{l_i}^{r_i} \alpha_j^* e^{-(x-\mu_j^*)^2/2} - \widehat{\alpha}_j e^{-(x-\widehat{\mu}_j)^2/2} dx + 1, && (13)\\
&= \frac{1}{\sqrt{2\pi}} \sum_{i=1,2} \sum_{j=1,2} \int_{l_i}^{r_i} F(\alpha^*, \widehat{\alpha}, \mu^*, \widehat{\mu}, x) dx + 1, && (14)
\end{aligned}
$$

We next examine the partial derivatives of L in order to understand the first order dynamics. We start with $\frac{\partial}{\partial l_i} L$:

$$
\frac{\partial}{\partial l_i} L = \frac{\partial}{\partial l_i} \left(\frac{1}{\sqrt{2\pi}} \sum_{i=1,2} \sum_{j=1,2} \int_{l_i}^{r_i} \alpha_j^* e^{-(x-\mu_j^*)^2/2} - \widehat{\alpha}_j e^{-(x-\widehat{\mu}_j)^2/2} dx + 1 \right) \tag{15}
$$

Which by Leibniz integral rule,

$$
\begin{aligned}
&= -\frac{1}{\sqrt{2\pi}} \sum_{j=1,2} \alpha_j^* e^{-(l_i-\mu_j^*)^2/2} - \widehat{\alpha}_j e^{-(l_i-\widehat{\mu}_j)^2/2} && (16)\\
&= -F(\alpha^*, \widehat{\alpha}, \mu^*, \widehat{\mu}, l_i) && (17)
\end{aligned}
$$

Similarly for $\frac{\partial}{\partial r_i} L$:

$$
\frac{\partial}{\partial r_i} L = \frac{\partial}{\partial r_i} \left(\frac{1}{\sqrt{2\pi}} \sum_{i=1,2} \sum_{j=1,2} \int_{l_i}^{r_i} \alpha_j^* e^{-(x-\mu_j^*)^2/2} - \widehat{\alpha}_j e^{-(x-\widehat{\mu}_j)^2/2} dx + 1 \right) \tag{18}
$$

Which by Leibniz integral rule,

$$
\begin{aligned}
&= \frac{1}{\sqrt{2\pi}} \sum_{j=1,2} \alpha_j^* e^{-(r_i-\mu_j^*)^2/2} - \widehat{\alpha}_j e^{-(r_i-\widehat{\mu}_j)^2/2} && (19)\\
&= F(\alpha^*, \widehat{\alpha}, \mu^*, \widehat{\mu}, r_i) && (20)
\end{aligned}
$$

Next, $\frac{\partial}{\partial \widehat{\mu}_j} L$ is a little more involved:

$$
\frac{\partial}{\partial \widehat{\mu}_j} L = \frac{\partial}{\partial \widehat{\mu}_j} \left(\frac{1}{\sqrt{2\pi}} \sum_{i=1,2} \sum_{j=1,2} \int_{l_i}^{r_i} \alpha_j^* e^{-(x-\mu_j^*)^2/2} - \widehat{\alpha}_j e^{-(x-\widehat{\mu}_j)^2/2} dx + 1 \right) \tag{21}
$$

Which by Leibniz integral rule,

$$
= \widehat{\alpha}_j \frac{1}{\sqrt{2\pi}} \sum_{i=1,2} e^{-(r_i-\widehat{\mu}_j)^2/2} - e^{-(l_i-\widehat{\mu}_j)^2/2} \tag{22}
$$

Finally, in the case of learning $\widehat{\alpha}$ by first order methods it is important to calculate the loss gradient with respect to α, $\frac{\partial}{\partial \widehat{\alpha}_j} L$.

$$\frac{\partial}{\partial \widehat{\alpha}_j} L = \frac{\partial}{\partial \widehat{\alpha}_j} \left(\frac{1}{\sqrt{2\pi}} \sum_{i=1,2} \sum_{j=1,2} \int_{l_i}^{r_i} \alpha_j^* e^{-(x-\mu_j^*)^2/2} - \widehat{\alpha}_j e^{-(x-\widehat{\mu}_j)^2/2} dx + 1 \right) \tag{23}$$

$$= -\frac{1}{\sqrt{2\pi}} \sum_{i=1,2} \int_{l_i}^{r_i} e^{-(x-\widehat{\mu}_j)^2/2} dx \tag{24}$$

Note that most of these can easily be formulated in terms of the probability distribution function (pdf) or cumulative distribution function (cdf) of a standard normal distribution.

Implementation Discoveries While implementing Li et al. we discovered a few sticking points. The most major was how to actually implement an efficient optimal discriminator. Essentially, finding the optimal discriminator values boils down to finding the zeros of the loss function. With two gaussians (even with $\alpha \neq \frac{1}{2}$), the loss function has at most 3 zeros. The problem is how to search and find all three of the zeros not knowing anything else about the loss function. This is far from optimal. Currently, we use a brentq function to find zeros in between any of the 4 μ values $\widehat{\mu}_1, \widehat{\mu}_2, \mu_1^*, \mu_2^*$, and if there are any zeros within distance 1 of the minimum and maximimum μ values. Empirically, this seems to work well enough for the cases we have tried (i.e. all convergence probabilities of the optimal discriminator are exactly 1, not near to 1), but we cannot prove that this procedure finds all of the zeros of the function, and in fact does not in some cases.