GAN Connoisseur: Can GANs Learn Simple 1D Parametric Distributions?

Manzil Zaheer*, **Chun-Liang Li***, **Barnabás Póczos**, **Ruslan Salakhutdinov**
Carnegie Mellon University
{manzilz, chunlial,bapoczos,rsalakhu}@cs.cmu.edu
(* denotes equal contribution)

Abstract

Generative Adversarial Network (GAN) has been shown to possess the capability to learn distributions of data, given infinite capacity of models [1, 2]. Empirically, approximations with deep neural networks seem to have "sufficiently large" capacity and lead to several success in many applications, such as image generation. However, most of the results are difficult to evaluate because of the curse of dimensionality and the unknown distribution of the data. To evaluate GANs, in this paper, we consider simple one-dimensional data coming from parametric distributions circumventing the aforementioned problems. We formulate rigorous techniques for evaluation under this setting. Based on this evaluation, we find that many state-of-the-art GANs are very difficult to train to learn the true distribution and can usually only find some of the modes. If the GAN has learned, such as MMD GAN, we observe it has some generalization capabilities.

1 Introduction

Generative Adversarial Network (GAN) is family of generative models that aims to generate novel samples from the data distribution rather than estimating the underlying distribution. Many existing works have demonstrated "performance" of GANs to be promising. For instance, a deep convolutional GAN [3] could generate variety of plausible natural images.

However, objectively and quantitatively evaluating a given GAN is still an open problem. Due to the curse of dimensionality, most non-parametric evaluations based on samples from the distributions are inapplicable. Most of existing works resort to heuristics [4], auxiliary classifiers [5], or Gaussian smoothing [6] in an attempt to justify working of existing GANs. Unfortunately, error analysis of such methods are difficult and as a result they are of limited value in understanding the true performance of GAN. In a different line of work, limited capacities of GANs have been shown by constructing birthday paradoxes [7]. Although this evaluation is theoretically grounded, it still relies on (subjective) human eyes to determine the duplicate samples (images).

In this paper, instead of studying GANs on standard benchmarks (high dimensional), we evaluate GANs on "seemly naive" one dimensional parametric distributions for both unconditional and conditional cases. We argue for this 1D experiment as it allows us to use many statistical tools, which might not be feasible in high dimensional cases, in order to gain several insights about GANs. We begin by deriving the analytical probability transformations in the 1D case and prove its uniqueness, which allows us to carry out many quantitative comparisons of the generator. Based on these comparison metrics, to our surprise, we find that representative GANs may fail on these simple cases and may be only able to capture (some of) the modes. Our experiments suggest us to rethink the capacity and ability of GANs and the what the property of the real-world data the GANs can learn reasonably well. On the other hand, if the GAN has learned the distribution well, we also observe its generalization capability.

2 Existing Literature on GANs

We are interested in sampling from $\mathbb{P}_{\mathcal{X}}$, where we are given $\{x_i\}_{i=1}^n \subset \mathcal{X}$ and $x_i \sim \mathbb{P}_{\mathcal{X}}$. Generative Adversarial Network (GAN) [1] trains a generator g_θ parametrized by θ to transform samples

31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA.

$z \sim \mathbb{P}_{\mathcal{Z}}$, where $z \in \mathcal{Z}$, into $g_\theta(z) \sim \mathbb{P}_\theta$ such that $\mathbb{P}_\theta \approx \mathbb{P}_{\mathcal{X}}$. During the training, we estimate the probabilistic discrepancy $d(\mathbb{P}_{\mathcal{X}}, \mathbb{P}_\theta)$, which is non-negative and $d(\mathbb{P}_{\mathcal{X}}, \mathbb{P}_\theta) = 0$ iff $\mathbb{P}_{\mathcal{X}} = \mathbb{P}_\theta$. We then update θ to minimize $d(\mathbb{P}_{\mathcal{X}}, \mathbb{P}_\theta)$ toward 0. Different $d(\mathbb{P}_{\mathcal{X}}, \mathbb{P}_\theta)$ results in different GAN works [1, 8, 9, 10, 11, 2, 12, 13, 14]. We briefly review some works here.

Vanilla GAN The pioneering work [1] proposed to train a simple binary classifier f_ϕ, which is called *Discriminator* (critics), to distinguish $\mathbb{P}_{\mathcal{X}}$ and $\mathbb{P}_\theta$. They formulate the following min-max objective to train f_ϕ and g_θ jointly,

$$\min_\theta \max_\phi \mathbb{E}_{x\sim\mathbb{P}_{\mathcal{X}}} \log f_\phi(x) + \mathbb{E}_{z\sim\mathbb{P}_{\mathcal{Z}}} \log(1 - f_\phi(g_\theta(z))). \tag{1}$$

This mini-max objective can be interpreted as a two-player game, where the generator g_θ tries to confuse the learned classifier f_ϕ [1]. The underlying $d(\mathbb{P}_{\mathcal{X}}, \mathbb{P}_\theta)$ for this setting was shown to be the Jensen-Shannon (JS) divergence [1]. Although [3] demonstrate good generated images on several benchmarks, training (vanilla) GAN by modeling f_ϕ as a binary classifier (JS divergence) is difficult and unstable due to its discontinuous nature [15].

Wasserstein GAN The use of Wasserstein distance $w(\mathbb{P}_{\mathcal{X}}, \mathbb{P}_\theta)$ between two probability distributions as $d(\mathbb{P}_{\mathcal{X}}, \mathbb{P}_\theta)$ in GAN was proposed by [2] and hence the name. Injecting the dual form of Wasserstein distance, the objective for WGAN training is

$$\min_\theta \underbrace{\sup_{\|f\|_L \le 1} \mathbb{E}_{x\sim\mathbb{P}_{\mathcal{X}}} f(x) - \mathbb{E}_{z\sim\mathbb{P}_{\mathcal{Z}}} f(g_\theta(z))}_{\text{Wasserstein distance: } w(\mathbb{P}_{\mathcal{X}}, \mathbb{P}_\theta),} \tag{2}$$

where $\|f\|_L \le 1$ is the set of functions whose Lipschitz constant is no larger than 1. Deep neural networks f_ϕ have been used to approximate $\|f\|_L \le 1$ and it is found that in many cases training WGAN is stabler than vanilla GAN [2].

MMD GAN Instead of using an auxiliary f_ϕ to measure $d(\mathbb{P}_{\mathcal{X}}, \mathbb{P}_\theta)$ as GAN and WGAN, use of kernel maximum mean discrepancy (MMD) $M_k(\mathbb{P}_{\mathcal{X}}, \mathbb{P}_\theta)$ [16] as $d(\mathbb{P}_{\mathcal{X}}, \mathbb{P}_\theta)$ has also be explored [8, 9]. The resulting method, called Generative Moment Matching Network (GMMN), uses the objective with a given kernel (e.g. Gaussian Kernel) k,

$$\min_\theta \underbrace{\mathbb{E}_{\mathbb{P}_{\mathcal{X}}} k(x, x') - 2\mathbb{E}_{\mathbb{P}_{\mathcal{X}}, \mathbb{P}_\theta} k(x, g_\theta(z)) + \mathbb{E}_{\mathbb{P}_\theta}, k(g_\theta(z), g_\theta(z'))}_{\text{MMD distance: } M_k(\mathbb{P}_{\mathcal{X}}, \mathbb{P}_\theta)}. \tag{3}$$

From (3), MMD distance $M_k(\mathbb{P}_{\mathcal{X}}, \mathbb{P}_\theta)$ can be easily estimated based on kernel evaluations without training any deep neural network. However, although several theoretical guarantees are shown in [16], GMMN fails on challenging benchmarks, such as CIFAR10. As a remedy, [12] propose *MMD GAN* that improves GMMN by considering a adversarially-learned kernel for MMD distance. The objective again becomes a min-max one and as derived by [12] is given by

$$\min_\theta \underbrace{\max_\phi \mathbb{E}_{\mathbb{P}_{\mathcal{X}}} k \circ f_\phi(x, x') - 2\mathbb{E}_{\mathbb{P}_{\mathcal{X}}, \mathbb{P}_\theta} k \circ f_\phi(x, g_\theta(z)) + \mathbb{E}_{\mathbb{P}_\theta}, k \circ f_\phi(g_\theta(z), g_\theta(z'))}_{\text{MMD GAN distance: } \max_\phi M_k \circ f_\phi(\mathbb{P}_{\mathcal{X}}, \mathbb{P}_\theta)} \tag{4}$$

where $k \circ f_\phi(x, x') = k(f_\phi(x), f_\phi(x'))$ and k is Gaussian kernel. Note that WGAN can be treated as a special case of MMD GAN by using linear kernel k, which only matches first-order moments.

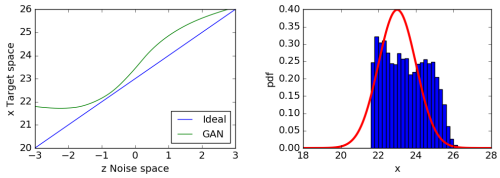
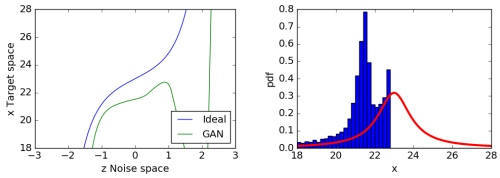
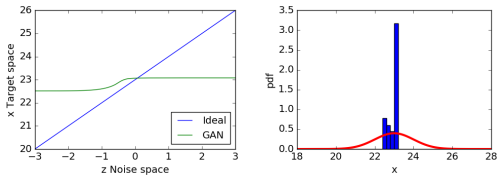
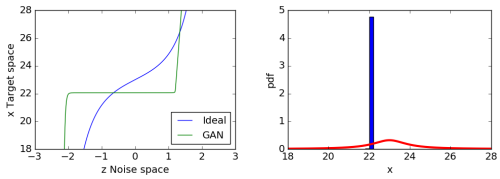
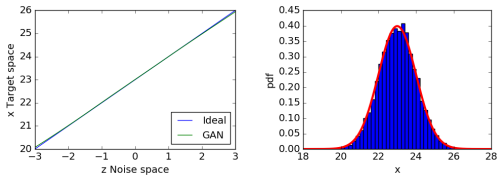
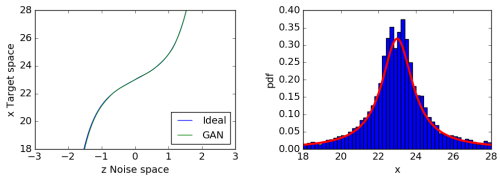
Other GAN works There are other GAN works considering different distance measures. We also review some works here. [10] generalize vanilla GAN [1] to consider general f-divergence. [11] match distributions by using Stein metric with score functions. [17] extend WGAN from mean matching to mean and covariance matching. [13] propose a WGAN-like algorithm by constraining the second-order moment, which recovers chi-squared distance. [14] use Cramer distance, which is similar to MMD GAN but with a different kernel.

3 Learning 1-D Distributions with GANs

We study GAN [1], WGAN [2] and MMD GAN [12][1] to transform $N(0, 1)$ and Unif$(-1, 1)$ to $N(23, 1)$ and Unif$(22, 24)$. For critics, we use 4-layer MLP with 11, 29, 11 and 1 units. For generators, we use use 4-layer MLP with 7, 13, 7 and 1 units. For both networks, we use ELU as the non-linear activation function. The learning rate are searched in $\{10^{-2}, 10^{-3}, 10^{-4}\}$ for Adam. The ratio of

[1] Note that MMD GAN can be treated as a kernelized version of WGAN.

Table 1: Performance of transforming $N(0,1)$ to different distributions.

Algo.	$N(23,1)$	Cauchy$(23,1)$
GAN	KSD: 0.2122 MAE: 0.5044 MSE: 0.3304 	KSD: 0.3571 MAE: 4.4009 MSE: 98.097 
WGAN	KSD: 0.4462 MAE: 0.6420 MSE: 0.6733 	KSD: 0.3495 MAE: 2.5635 MSE: 56.390 
MMD GAN	KSD: 7.50e-3 MAE: 8.89e-3 MSE: 8.96e-5 	KSD: 0.0123 MAE: 0.4099 MSE: 5.5917 

updating critics and generators are searched in $\{2{:}1, 3{:}1, 4{:}1, 5{:}1\}$. We train each setting for 1000000 iterations and report the best results for each algorithm.

Theorem 1. *(Uniqueness of 1D transformation) Given two one dimensional random variable $Z \sim \mathbb{P}_{\mathcal{Z}}$ and $X \sim \mathbb{P}_{\mathcal{X}}$ where both the probability distributions $\mathbb{P}_{\mathcal{Z}}$ and $\mathbb{P}_{\mathcal{Z}}$ are twice differentiable and have support almost everywhere, there are at most two differentiable f such that $f(Z) = X$.*

Proof. Let $(\Omega, \mathcal{B})$ be a measurable space and $\mathbb{P}_{\mathcal{Z}}$ and $\mathbb{P}_{\mathcal{X}}$ be two measures on this space. Denote by $p_Z(\cdot)$ and $p_X(\cdot)$ be the densities corresponding to the measures $\mathbb{P}_{\mathcal{Z}}$ and $\mathbb{P}_{\mathcal{X}}$ respectively. Using change of measure, one can write,

$$\begin{aligned} \forall A \in \mathcal{B}: \quad & \int_A p_X(x) \left|\frac{dx}{dz}\right| dz = \int_A p_Z(z) dz \\ \forall A \in \mathcal{B}: \quad & \int_A \left(p_X(x) \left|\frac{dx}{dz}\right| - p_Z(z) \right) dz = 0 \end{aligned} \tag{5}$$

Since the integral is 0 for any set A, the integrand must be zero identically, i.e. we have the following differential equation:

$$p_X(x) \left|\frac{dx}{dz}\right| - p_Z(z) = 0 \tag{6}$$

This can be split as two first-order separable ordinary differential equations (ODE):

$$p_X(x)\frac{dx}{dz} - p_Z(z) = 0 \quad \text{or} \quad p_X(x)\frac{dx}{dz} + p_Z(z) = 0 \tag{7}$$

Now that solving the first ODE for example leads to $\mathbb{P}_{\mathcal{X}}(x) = \mathbb{P}_{\mathcal{Z}}(z) + c$, but as $x, z \to \infty$ we know that $\mathbb{P}_{\mathcal{X}}(x), \mathbb{P}_{\mathcal{Z}}(z) \to 1$, as a result $c = 0$ is the only valid option. Similar reasoning works for the other case. As we assumed $\forall x : p_X(x) > 0$ and $\forall z : p_Z(z) > 0$, then each the ODE in (7) along with the boundary condition has one unique solution, which completes the proof. □

Based on Theorem 1, there are at most two continuous probability transformations f for 1D distribution, which can be simply derived by *probability integral transform*. We show the results of transforming $\mathbb{P}_{\mathcal{Z}} = N(0,1)$ to $N(23,1)$ and Cauchy$(23,1)$ in Table 1. We compare (1) $g_\theta(z)$ with the true transformation function and (2) the empirical distribution from $10,000$ samples with $\mathbb{P}_{\mathcal{X}}$. We

also report *KSD*, *MAE* and *MSE*, which are defined as

$$\text{KSD} = \sup_x |\mathbb{P}_{\mathcal{X}}(x) - \mathbb{P}_\theta(x)|, \ \text{MAE} = \int_{-\infty}^{\infty} |f(z) - \hat{f}(z)| d\mathbb{P}_{\mathcal{Z}}(z), \ \text{MSE} = \int_{-\infty}^{\infty} (f(z) - \hat{f}(z))^2 d\mathbb{P}_{\mathcal{Z}}(z),$$

where KSD is the metric used for Kolmogorov-Smirnov goodness of fit test.

As can be seen from Table 1 GAN fails to learn either distribution, which may be caused by the difficulty of its training [15]. WGAN can only match the mode of $\mathbb{P}_{\mathcal{X}}$. It is consistent with (2), which only compares the first-order moment of the transformed data $f(x)$. Without the large capacity of network to approximate Lipschitz functions well, WGAN fails to match the distributions. Here we used weight-clipping as adopted in [2]. However, the more advanced WGAN-GP [18] results in similar performance. We use the implementation provided by [18][2] to learn a 2D Gaussian distribution with identity covariance, i.e. unit variance in each of the two directions. 4-layer MLP with 512 hidden units for both critic and generator are adopted by [18]. In Figure 1, the result is similar to Table 1 that WGAN cannot learn the distribution well. On the other hand, MMD GAN, which can be treated as a kernelized extension of WGAN, leverages the higher-order moment matching of MMD, and successfully matches distributions. If we train GAN/WGAN with much deeper and wider networks, the performance is improved but is still not comparable to MMD GAN.

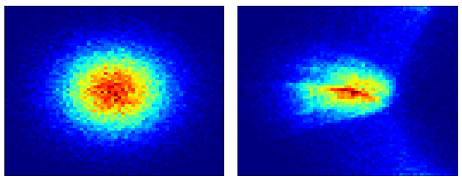


Figure 1: Truth vs learned from WGAN-GP code on Gaussian with $\mu = 0$ and $\Sigma = I$.

3.1 1D Distribution with Conditional GAN

We extend the 1D generation experiment to the conditional version based on [20]. In addition to $z \sim \text{Unif}(-1, 1)$, the generator is also given μ_i, with the goal to generate the samples $g_\theta(z|\mu_i)$ from $N(\mu_i, 1)$. The critics then measure the discrepancy between $\mathbb{P}(x|\mu)$ and $\mathbb{P}(g_\theta(z|\mu))$. In the training, μ_i is sampled from $[-10, -8, \cdots, 8, 10]$ uniformly. It can be seen from Figure 2, similar to the previous result, both GAN and WGAN failed to learn the conditional distributions. In the training, we only sampled μ among integers in $[-10, 10]$, but a well learned $g(z|\mu_i)$, like the one obtained from MMD GAN, generalizes well with non-integer μ as well as the μ outside of the range. This means without any explicit supervision, GAN could learn the ideal transformation of $X = \mu + Z$. Note that the generator has more capacity (4-layer MLP) than to learn just a straight line. It supports GAN has capabilities to generalize and learn the distribution in a different perspective from [7].

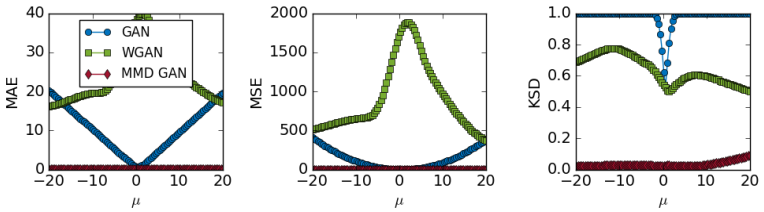


Figure 2: The result of conditional generation on μ.

4 Discussion

In this paper, we study GANs' learning capability on simple one-dimensional parametric distributions. Surprisingly, vanilla GAN and WGAN fail on learning these distributions but only capture the mode of the distributions. MMD GAN, which leverages prior knowledge on two-sample test, seems to be able to learn. It may suggest a future direction of designing GAN's algorithm by incorporating more existing statistics results. In addition, we should rethink the property of current benchmarks, which GAN and WGAN seem to have promising performance on them. On the other hand, we also study the conditional generation on 1D parametric distribution. We show that, if GAN has learned the distributions, it possesses some generalization capability rather than just memorizing the data.

[2]It is used to learn 8 mixture of Gaussians, where the covariance of each Gaussian is $0.0003125 \times I$ in line with the setup used by many literature on GANs [4, 18, 19]. However, we feel the setup is not good for evaluating the performance of GANs as at such small variances the distribution essentially becomes point mass. The success shown by previous works in capturing this almost point mass like GMM is in accordance to our observation that GAN/WGAN is only able to identify the modes of the distribution but not the shape.

References

[1] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron C. Courville, and Yoshua Bengio. Generative adversarial nets. In *NIPS*, 2014.

[2] Martín Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein GAN. In *ICML*, 2017.

[3] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. In *ICLR*, 2016.

[4] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. In *NIPS*, pages 2226–2234, 2016.

[5] David Lopez-Paz and Maxime Oquab. Revisiting classifier two-sample tests. In *ICLR*, 2017.

[6] Yuhuai Wu, Yuri Burda, Ruslan Salakhutdinov, and Roger Grosse. On the quantitative analysis of decoder-based generative models. In *ICLR*, 2017.

[7] Sanjeev Arora and Yi Zhang. Do gans actually learn the distribution? an empirical study. *arXiv preprint arXiv:1706.08224*, 2017.

[8] Yujia Li, Kevin Swersky, and Richard Zemel. Generative moment matching networks. In *ICML*, 2015.

[9] Gintare Karolina Dziugaite, Daniel M. Roy, and Zoubin Ghahramani. Training generative neural networks via maximum mean discrepancy optimization. In *UAI*, 2015.

[10] Sebastian Nowozin, Botond Cseke, and Ryota Tomioka. f-gan: Training generative neural samplers using variational divergence minimization. In *NIPS*, 2016.

[11] Dilin Wang and Qiang Liu. Learning to draw samples: With application to amortized MLE for generative adversarial learning. *CoRR*, abs/1611.01722, 2016.

[12] Chun-Liang Li, Wei-Cheng Chang, Yu Cheng, Yiming Yang, and Barnabás Póczos. MMD GAN: towards deeper understanding of moment matching network. In *NIPS*, 2017.

[13] Youssef Mroueh and Tom Sercu. Fisher gan. In *NIPS*, 2017.

[14] Marc G. Bellemare, Ivo Danihelka, Will Dabney, Shakir Mohamed, Balaji Lakshminarayanan, Stephan Hoyer, and Rémi Munos. The cramer distance as a solution to biased wasserstein gradients. *CoRR*, abs/1705.10743, 2017.

[15] Martin Arjovsky and Léon Bottou. Towards principled methods for training generative adversarial networks. In *ICLR*, 2017.

[16] Arthur Gretton, Karsten M. Borgwardt, Malte J. Rasch, Bernhard Schölkopf, and Alexander Smola. A kernel two-sample test. *JMLR*, 2012.

[17] Youssef Mroueh, Tom Sercu, and Vaibhava Goel. McGan: Mean and covariance feature matching GAN. In *ICML*, 2017.

[18] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron Courville. Improved training of wasserstein gans. In *NIPS*, 2017.

[19] Vaishnavh Nagarajan and J Zico Kolter. Gradient descent gan optimization is locally stable. In *NIPS*, 2017.

[20] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*, 2014.

Table 1: Performance of Various GAN

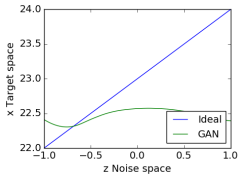
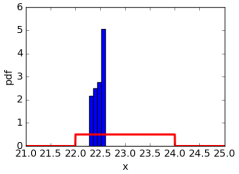
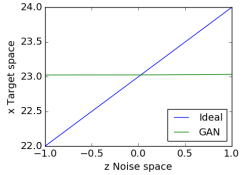
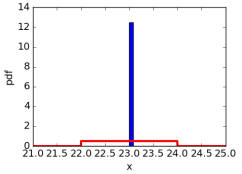
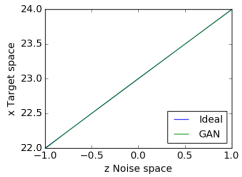
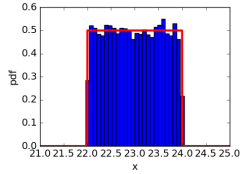
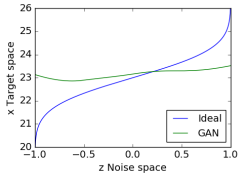
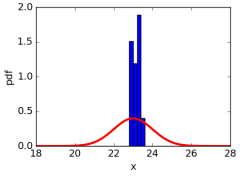
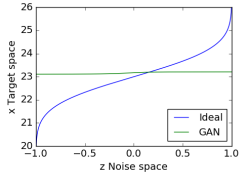
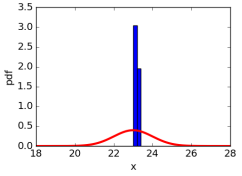
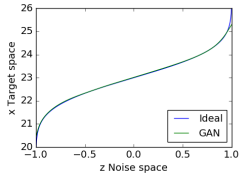
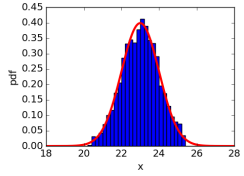
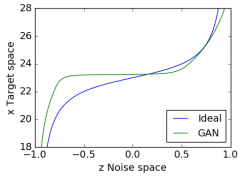
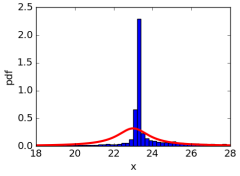
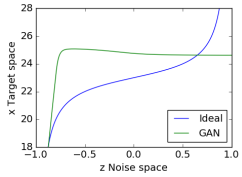
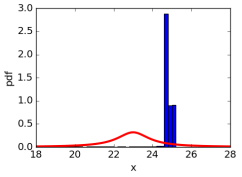
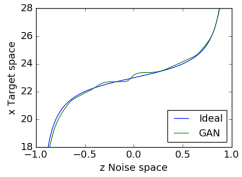
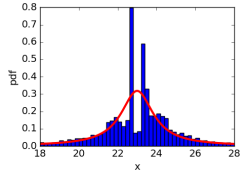
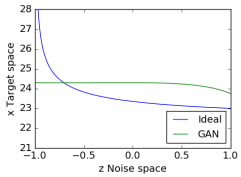
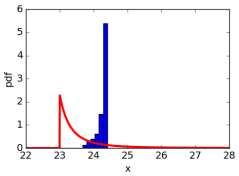
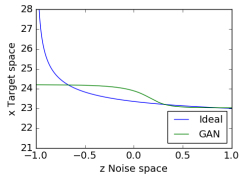
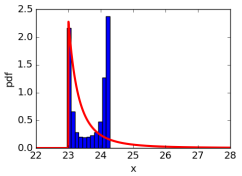
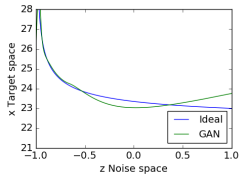
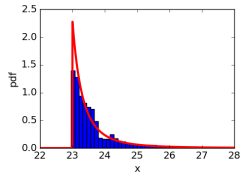
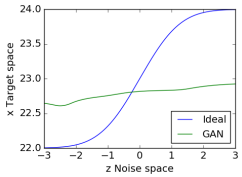
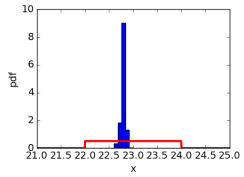
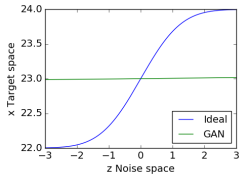
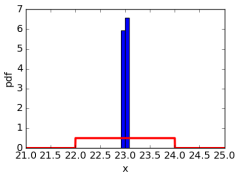
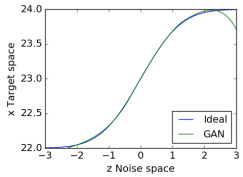
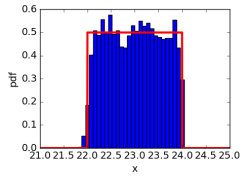
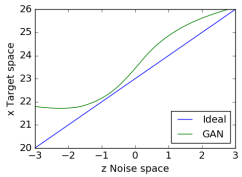
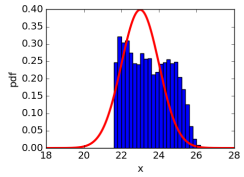
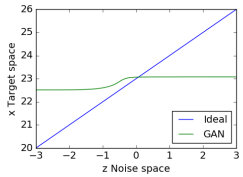
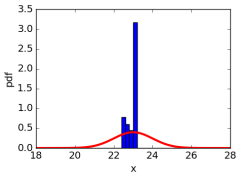
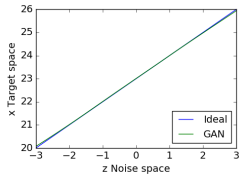
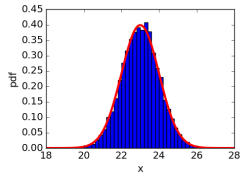
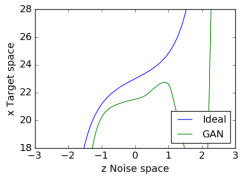
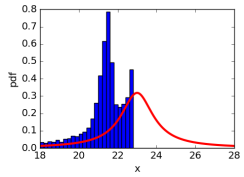
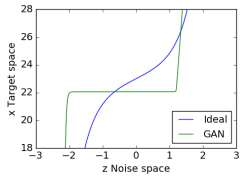
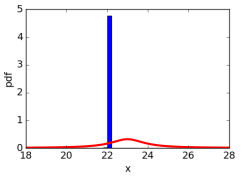
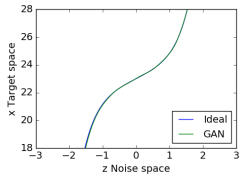
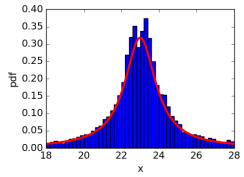
	Target	Simple		WGAN		MMD GAN	
Uniform	Uniform	MAE: 0.58596	MSE: 0.54683	MAE: 0.49854	MSE: 0.33143	MAE: 0.018418	MSE: 0.00045452
							
	Gaussian	MAE: 0.64727	MSE: 0.73553	MAE: 0.77169	MSE: 0.95643	MAE: 0.023283	MSE: 0.0023875
							
	Cauchy	MAE: 3.1459	MSE: 584.39	MAE: 4.1899	MSE: 613.05	MAE: 1.573	MSE: 376.41
							
	Pareto	MAE: 10.271	MSE: 2.0718e+06	MAE: 9.7667	MSE: 2.072e+06	MAE: 9.6763	MSE: 2.0718e+06
							

Table 2: Performance of Various GAN

	Target	Simple		WGAN		MMD GAN	
Gaussian	Uniform	MAE: 0.48152	MSE: 0.3226	MAE: 0.49312	MSE: 0.32463	MAE: 0.029086	MSE: 0.0010261
							
	Gaussian	MAE: 0.5044	MSE: 0.33039	MAE: 0.64197	MSE: 0.67326	MAE: 0.0088905	MSE: 8.9634e-05
							
	Cauchy	MAE: 4.4009	MSE: 98.097	MAE: 2.5635	MSE: 56.39	MAE: 0.5397	MSE: 25.591
							
	Pareto	MAE: 0.72926	MSE: 1.0338	MAE: 0.48144	MSE: 0.95841	MAE: 0.25368	MSE: 0.21395