A New Paradigm for Generative Adversarial Networks Based on Randomized Decision Rules

Purdue University

Supplementary Material

This supplementary material is organized as follows. Section S1 gives the proofs for the theoretical results of the paper. Section S2 defines the metrics, including the inception score, Wasserstein distance, and maximum mean discrepancy, used for quantifying the performance of image generation for different methods. Section S3 presents more numerical examples, including those on image generation, conditional independence tests, and nonparametric clustering. Section S4 presents parameter settings used in the numerical experiments.

S1 Theoretical Proofs

S1.1 Proof of Lemma 1

Proof.

$$\begin{aligned}\mathbb{E}_{\pi_g}\mathcal{J}_d(\theta_d;\theta_g) &= \int \phi_1(D_{\theta_d}(x))p_{data}(x)dx + \int\int \phi_2(D_{\theta_d}(G_{\theta_g}(z)))q(z)\pi_g(\theta_g)dzd\theta_g \\ &= \int \phi_1(D_{\theta_d}(x))p_{data}(x)dx + \int\int \phi_2(D_{\theta_d}(x))p_{\theta_g}(x)\pi_g(\theta_g)d\theta_g dx \qquad \text{(S1.1)} \\ &= \int \left[\phi_1(D_{\theta_d}(x))p_{data}(x) + \phi_2(D_{\theta_d}(x))p_{\pi_g}(x)\right]dx,\end{aligned}$$

where the mixture generator formed by π_g can be viewed as a single super

generator θ_g^* such that $p_{\theta_g^*}(x) = p_{\pi_g}(x)$. Then, by the proof of Theorem 1 of Goodfellow et al. (2014), we have $\min_{\pi_g} \max_{\theta_d} \mathbb{E}_{\pi_g} \mathcal{J}_d(\theta_d; \theta_g) = -\log(4)$. It is easy to verify that at the Nash equilibrium point, $\mathbb{E}_{\tilde{\pi}_g} \mathcal{J}_d(\tilde{\theta}_d; \theta_g) = -\log(4)$.

By the proof of Theorem 1 of Goodfellow et al. (2014), if

$$\tilde{\theta}_d = \arg\max_{\theta_d} \mathbb{E}_{\tilde{\pi}_g} \mathcal{J}_d(\theta_d; \theta_g)$$

holds, then $\mathbb{E}_{\tilde{\pi}_g} \mathcal{J}_d(\tilde{\theta}_d; \theta_g) = -\log(4)$ implies the Jensen–Shannon divergence $JSD(p_{data}|p_{\tilde{\pi}_g}) = 0$ and thus $p_{\tilde{\pi}_g} = p_{data}$. Further, by Proposition 1 of Goodfellow et al. (2014), we have $D_{\tilde{\theta}_d}(x) = 1/2$ when $p_{\tilde{\pi}_g} = p_{data}$ holds. □

S1.2 Proof of Theorem 1

Proof. The proof consists of two steps. First, we would prove that

$$\int \mathcal{J}_d(\tilde{\theta}_d; \theta_g) \pi(\theta_g | \tilde{\theta}_d, \mathcal{D}) d\theta_g = -\log 4, \quad \text{as } N \to \infty. \tag{S1.2}$$

For the game (2.4), it is easy to see that

$$\begin{aligned}
&\min_{\pi_g} \max_{\theta_d} \mathbb{E}_{\theta_g \sim \pi_g} \mathcal{J}_d(\theta_g; \theta_g) \le \max_{\theta_d} \int \mathcal{J}_d(\theta_d; \theta_g) \pi(\theta_g | \theta_d, \mathcal{D}) d\theta_g = \int \mathcal{J}_d(\tilde{\theta}_d; \theta_g) \pi(\theta_g | \tilde{\theta}_d, \mathcal{D}) d\theta_g \\
&= -\log 4 + \frac{1}{N} \int \{N(\mathcal{J}_d(\tilde{\theta}_d; \theta_g) + \log 4) - \log q_g(\theta_g) + \log Z(\tilde{\theta}_d)\} \pi(\theta_g | \tilde{\theta}_d, \mathcal{D}) d\theta_g \\
&\quad + \frac{1}{N} \int \{\log q_g(\theta_g)\} \pi(\theta_g | \tilde{\theta}_d, \mathcal{D}) d\theta_g - \frac{1}{N} \log Z(\tilde{\theta}_d) \\
&= -\log 4 + (I) + (II) + (III),
\end{aligned} \tag{S1.3}$$

where $Z(\tilde{\theta}_d)$ is the normalizing constant of $\pi(\theta_g|\tilde{\theta}_d, \mathcal{D})$.

As implied by (S1.1), $\max_{\theta_d} \int \mathcal{J}_d(\theta_d; \theta_g)\pi_g(\theta_g)d\theta_g$ is equivalent to $\max_{\theta_d} \mathcal{J}_d(\theta_d; \theta_g)$ for a fixed generator θ_g that $p_{\theta_g}(x) = p_{\pi_g}(x)$ holds. Therefore, by Theorem 1 of Goodfellow et al. (2014), we have $\mathcal{J}_d(\tilde{\theta}_d; \theta_g) \geq -\log 4$ for any $\theta_g \in \Theta_g$. That is, $N(\mathcal{J}_d(\tilde{\theta}_d; \theta_g) + \log 4) - \log q_g(\theta_g)$ can be treated as the energy of the posterior $\pi(\theta_g|\tilde{\theta}_d, \mathcal{D})$, and then

$$(I) = -\frac{1}{N}\int \{\log \pi(\theta_g|\tilde{\theta}_d, \mathcal{D})\}\pi(\theta_g|\tilde{\theta}_d, \mathcal{D})d\theta_g.$$

By the Kullback-Leibler divergence $D_{KL}(\pi(\theta_g|\tilde{\theta}_d, \mathcal{D})|q_g) \geq 0$,

$$(II) \leq \frac{1}{N}\int \{\log \pi(\theta_g|\tilde{\theta}_d, \mathcal{D})\}\pi(\theta_g|\tilde{\theta}_d, \mathcal{D})d\theta_g.$$

As justified in Remark S1, $|\log Z(\tilde{\theta}_d)|$ is of the order $O(dim(\theta_g)\log N)$ and thus $(III) \to 0$ as $N \to \infty$. Summarizing these terms, we have

$$\int \mathcal{J}_d(\tilde{\theta}_d; \theta_g)\pi(\theta_g|\tilde{\theta}_d, \mathcal{D})d\theta_g \overset{N\to\infty}{\leq} -\log 4. \tag{S1.4}$$

By (S1.3) and Lemma 1, we have

$$\int \mathcal{J}_d(\tilde{\theta}_d; \theta_g)\pi(\theta_g|\tilde{\theta}_d, \mathcal{D})d\theta_g \geq \min_{\pi_g}\max_{\theta_d} \mathbb{E}_{\theta_g\sim\pi_g}\mathcal{J}_d(\theta_g; \theta_g) = -\log 4.$$

Combining it with (S1.4), we can conclude equation (S1.2).

Next, to apply Lemma 1 to claim that $(\tilde{\theta}_d, p_{\tilde{\pi}_g})$ is a Nash equilibrium point, we still need to prove that $\tilde{\theta}_d$ is also the maximizer of $\int \mathcal{J}_d(\theta_d; \theta_g)\pi(\theta_g|\tilde{\theta}_d, \mathcal{D})d\theta_g$. We do this by proof of contradiction. Suppose

$$\|\tilde{\theta}_d - \arg\max_{\theta_d} \int \mathcal{J}_d(\theta_d; \theta_g)\pi(\theta_g|\tilde{\theta}_d, \mathcal{D})d\theta_g\| > \delta_0$$

for some $\delta_0 > 0$. Then, by Proposition 1 of Goodfellow et al. (2014), there exist a function $\epsilon(x)$ and a constant $\epsilon_0 > 0$ such that

$$D_{\tilde{\theta}_d}(x) = \frac{p_{data}(x) + \epsilon(x)}{p_{data}(x) + p_{\tilde{\pi}_g}(x)},$$

and $|\epsilon(x)| > \epsilon_0$ on some non-zero measure set of $\mathcal{X}$, where $\mathcal{X}$ denotes the domain of x and $-p_{data}(x) \leq \epsilon(x) \leq p_{\tilde{\pi}}(x)$ for ensuring $0 \leq D_{\tilde{\theta}_d}(x) \leq 1$. Following the proof of Theorem 1 of Goodfellow et al. (2014), we have

$$\begin{aligned}
&\int \mathcal{J}_d(\tilde{\theta}_d; \theta_g)\pi(\theta_g|\tilde{\theta}_d, \mathcal{D})d\theta_g = \mathbb{E}_{x\sim p_{data}} \log \frac{p_{data}(x) + \epsilon(x)}{p_{data}(x) + p_{\tilde{\pi}_g}(x)} + \mathbb{E}_{x\sim p_{\tilde{\pi}_g}} \log \frac{p_{\tilde{\pi}_g}(x) - \epsilon(x)}{p_{data}(x) + p_{\tilde{\pi}_g}(x)} \\
&= -\log 4 + 2JSD(p_{data}|p_{\tilde{\pi}_g}) + \mathbb{E}_{x\sim p_{data}} \log(1 + \frac{\epsilon(x)}{p_{data}(x)}) + \mathbb{E}_{x\sim p_{\tilde{\pi}_g}} \log(1 - \frac{\epsilon(x)}{p_{\tilde{\pi}_g}(x)}).
\end{aligned} \tag{S1.5}$$

If $p_{\tilde{\pi}_g} = p_{data}$, then $JSD(p_{data}|p_{\tilde{\pi}_g}) = 0$, $\mathbb{E}_{x\sim p_{data}} \log(1 + \frac{\epsilon(x)}{p_{data}(x)}) +$

$\mathbb{E}_{x\sim p_{\tilde{\pi}_g}} \log(1 - \frac{\epsilon(x)}{p_{\tilde{\pi}_g}(x)}) < 0$ by Jensen's inequality, and thus

$$\int \mathcal{J}_d(\tilde{\theta}_d; \theta_g)\pi(\theta_g|\tilde{\theta}_d, \mathcal{D})d\theta_g < -\log 4.$$

In what follows we show that this is in contradiction to (S1.2) by showing that the $\tilde{\theta}_d$ corresponding to $p_{\tilde{\pi}_g} = p_{data}$ is a solution to the problem $\max_{\theta_d} \int \mathcal{J}_d(\theta_d; \theta_g)\pi(\theta_g|\theta_d, \mathcal{D})d\theta_g$.

Suppose that N is sufficiently large and $p_{\pi_g} = p_{data}$ holds, then we have: (i) $D_{\tilde{\theta}'_d} = 1/2$ by (S1.1) and Proposition 1 of Goodfellow et al. (2014), where $\tilde{\theta}'_d = \arg\max \mathbb{E}_{\pi_g} \mathcal{J}_d(\theta_d; \theta_g)$ with $p_{\pi_g} = p_{data}$; (ii) in the space of p_{θ_g} the posterior $\pi(\theta_g|\tilde{\theta}'_d, \mathcal{D})$ has the mode at $p_{\theta_g} = p_{data}$ as $N \to \infty$ following from the arguments that $\mathcal{J}_g(\theta_g; \tilde{\theta}'_d)$ is concave with respect to p_{θ_g} as shown in Proposition 2 of Goodfellow et al. (2014), and that $\mathcal{J}_g(\theta_g; \tilde{\theta}'_d)$ attains its maximum at $p_{\theta_g} = p_{data}$ by Theorem 1 of Goodfellow et al. (2014); and (iii) $\mathcal{J}_d(\tilde{\theta}'_d; \theta_g) = -\log 4$ at the posterior mode $p_{\theta_g} = p_{data}$. Then, by Laplace approximation (Kass et al., 1990), we have $\int \mathcal{J}_d(\tilde{\theta}'_d; \theta_g)\pi(\theta_g|\tilde{\theta}'_d, \mathcal{D})d\theta_g \to -\log 4$ and $p_{\tilde{\pi}'_g} = \int p_{\theta_g}\pi(\theta_g|\tilde{\theta}'_d, \mathcal{D})d\theta_g = p_{data}$ as $N \to \infty$. That is, the $\tilde{\theta}_d$ corresponding to $p_{\tilde{\pi}_g} = p_{data}$ (changing the notations $\tilde{\theta}'_d$ to $\tilde{\theta}_d$ and $p_{\tilde{\pi}'_g}$ to $p_{\tilde{\pi}_g}$) is indeed a maximizer of $\int \mathcal{J}_d(\theta_d; \theta_g)\pi(\theta_g|\theta_d, \mathcal{D})d\theta_g$ as $N \to \infty$. Note that $\pi(\theta_g|\tilde{\theta}_d, \mathcal{D})$ may contain multiple equal modes in the space of θ_g due to the nonidentifiability of the neural network model, which does not affect the

validity of the above arguments. Therefore, by the contradiction, we can conclude that $\tilde{\theta}_d = \arg\max_{\theta_d} \int \mathcal{J}_d(\theta_d;\theta_g)\pi(\theta_g|\tilde{\theta}_d,\mathcal{D})d\theta_g$ by the arbitrariness of δ_0.

The proof can then be concluded by Lemma 1 with the results of the above two steps. □

Remark S1. The order of $|\log Z(\tilde{\theta}_d)|$ given in the proof of Theorem 1 can be justified based on Laplace approximation (Kass et al., 1990), and the justification can be extended to any fixed value of θ_d. Let $c = \min_{\theta_g\in\Theta_g} \mathcal{J}_d(\theta_d;\theta_g)$ for any fixed value of θ_d. Applying the Laplace approximation to the integral $\int \exp\{-N(\mathcal{J}_d(\theta_d;\theta_g)-c)\}q_g(\theta_g)d\theta_g$, we have

$$Z(\theta_d) = (2\pi)^{dim(\theta_g)/2}[det(N\boldsymbol{H}_e)]^{-1/2}\exp\{-N(\mathcal{J}_d(\theta_d;\hat{\theta}_g)-c)\}q_g(\hat{\theta}_g)\Big(1+O(\frac{1}{N})\Big), \tag{S1.6}$$

where $\hat{\theta}_g = \arg\max_{\theta_g\in\Theta_g}\{-(\mathcal{J}_d(\theta_d,\theta_g)-c)+\frac{1}{N}\log q_g(\theta_g)\}$, $\boldsymbol{H}_e$ is the Hessian of $\mathcal{J}_d(\theta_d;\theta_g)-c-\frac{1}{N}\log q_g(\theta_g)$ evaluated at $\hat{\theta}_g$, and $det(\cdot)$ denotes the determinant operator. By the convexity of $\mathcal{J}_d(\theta_d,\theta_g)$ (with respect to p_{θ_g} as shown in Proposition 2 of Goodfellow et al. (2014)) and the boundedness of the prior density function by Assumption (ii) of Theorem 1, it is easy to see that $N(\mathcal{J}_d(\theta_d,\hat{\theta}_g)-c)-\log q_g(\hat{\theta}_g)$ is finite and thus $(\mathcal{J}_d(\theta_d;\hat{\theta}_g)-c)-\frac{1}{N}\log q_g(\hat{\theta}_g) \to 0$ as $N\to\infty$. If all the eigenvalues of $\boldsymbol{H}_e$ are bounded by some positive

constants, then $-\frac{1}{N}\log Z(\theta_d) = O(dim(\theta_g)\log N/N) = o(1)$. Finally, we note that the analytical assumptions for Laplace's method (Kass et al., 1990) can be verified based on the convexity of $\mathcal{J}_d(\theta_d, \theta_g)$ and some mild assumptions on the derivatives of $\mathcal{J}_d(\theta_d, \theta_g) - c - \frac{1}{N}\log q_g(\theta_g)$ at $\hat{\theta}_g$; and that the posterior may contain multiple equal modes in the space of θ_g due to the nonidentifiability of the neural network model, which does not affect the validity of the above approximation.

S1.3 Proof of Corollary 1

Proof. Extension of Theorem 1 to the case $\phi_3(D) = \log(D)$ can be justified as follows. Let

$$\pi'(\theta_g|\tilde{\theta}_d, \mathcal{D}) = \exp\{N(-\mathbb{E}_{x\sim p_{data}}\phi_1(D_{\tilde{\theta}_d}(x)) + \mathbb{E}_{x\sim p_{\theta_g}}\phi_3(D_{\tilde{\theta}_d}(x))\}q_g(\theta_g)/Z'(\tilde{\theta}_d)$$

for $\phi_3(D) = \log(D)$, and let

$$\pi(\theta_g|\tilde{\theta}_d, \mathcal{D}) = \exp\{N(-\mathbb{E}_{x\sim p_{data}}\phi_1(D_{\tilde{\theta}_d}(x)) + \mathbb{E}_{x\sim p_{\theta_g}}\phi_3(D_{\tilde{\theta}_d}(x)) - c)\}q_g(\theta_g)/Z(\tilde{\theta}_d)$$

for $\phi_3(D) = -\log(1-D)$, where $c = -\log 4$, and $Z'(\tilde{\theta}_d)$ and $Z(\tilde{\theta}_d)$ denote their respective normalizing constants. Then

$$\begin{aligned}
&\int \mathcal{J}_d(\tilde{\theta}_d;\theta_g)\pi'(\theta_g|\tilde{\theta}_d,\mathcal{D}) = c + \frac{1}{N}\int \Big[N(\mathcal{J}_d(\tilde{\theta}_d;\theta_g) - c) - \log q_g(\theta_g) + \log Z(\tilde{\theta}_d)\Big]\,\pi'(\theta_g|\tilde{\theta}_d,\mathcal{D})d\theta_g \\
&\quad + \frac{1}{N}\int \log q_g(\theta_g)\pi'_g(\theta_g|\tilde{\theta}_d,\mathcal{D})d\theta_g - \frac{1}{N}\log Z(\tilde{\theta}_d) \\
&\le c + \frac{1}{N}\int \Big[-\log \pi(\theta_g|\tilde{\theta}_d,\mathcal{D}) + \log \pi'(\theta_g|\tilde{\theta}_d,\mathcal{D})\Big]\,\pi'(\theta_g|\tilde{\theta}_d,\mathcal{D})d\theta_g - \frac{1}{N}\log Z(\tilde{\theta}_d) \\
&= c + (I) + (II),
\end{aligned}$$

where the inequality follows from that the Kullback-Leibler divergence $D_{KL}(\pi'_g|q_g) \ge 0$.

By Remark S1, we have $(II) \to 0$ as $N \to \infty$. The term (I) is the Kullback-Leibler divergence between $\pi'(\theta_g|\tilde{\theta}_d,\mathcal{D})$ and $\pi(\theta_g|\tilde{\theta}_d,\mathcal{D})$. By the upper bound of the Kullback-Leibler divergence (Dragomir et al., 2000), we have

$$\begin{aligned}
(I) &\le \frac{1}{N}\int \frac{\pi'(\theta_g|\tilde{\theta}_d,\mathcal{D})}{\pi(\theta_g|\tilde{\theta}_d,\mathcal{D})}\pi'(\theta_g|\tilde{\theta}_d,\mathcal{D})d\theta_g - \frac{1}{N} \\
&= \frac{1}{N} \times \frac{Z(\tilde{\theta}_d)}{Z'(\tilde{\theta}_d)} \times \int \prod_{x_i \sim p_{\theta_g}, i=1,2,\ldots,N} [4D_{\tilde{\theta}_d}(x_i)(1 - D_{\tilde{\theta}_d}(x_i))]\pi'(\theta_g|\tilde{\theta}_d,\mathcal{D})d\theta_g - \frac{1}{N} \\
&= \frac{1}{N} \times (I_1) \times (I_2) - \frac{1}{N}.
\end{aligned}$$

Since $4D_{\tilde{\theta}_d}(x_i)(1 - D_{\tilde{\theta}_d}(x_i)) \le 1$ for each x_i, we have $(I_2) \le 1$. Next, we consider the term (I_1). For both choices of ϕ_3, as implied by (S1.1) where the mixture generator proposed in the paper is represented as a single super generator, the arguments in Goodfellow et al. (2014) on the non-saturating case can be applied here, and thus $\pi(\theta_g|\tilde{\theta}_d,\mathcal{D})$ and $\pi'(\theta_g|\tilde{\theta}_d,\mathcal{D})$ have the same maximum *a posteriori* (MAP) estimate $\hat{\theta}_g$ as $N \to \infty$. Further, by Lemma

1, we have $D_{\tilde{\theta}_d}(x) = 1/2$ for any $x \in p_{data}$. Then it is easy to see that $\log \pi(\theta_g|\tilde{\theta}_d, \mathcal{D})$ and $\log \pi'(\theta_g|\tilde{\theta}_d, \mathcal{D})$ have exactly the same first and second gradients at $(\tilde{\theta}_d, \hat{\theta}_g)$, which implies that they have the same Hessian matrix. Therefore, by (S1.6), $(I_1) = Z(\tilde{\theta}_d)/Z'(\tilde{\theta}_d) \to 1$ as $N \to \infty$. Summarizing (I_1) and (I_2), we have $(I) \to 0$ as $N \to \infty$. Summarizing all the above arguments, we have $\int \mathcal{J}_d(\tilde{\theta}_d; \theta_g)\pi'_g(\theta_g|\tilde{\theta}_d, \mathcal{D}) \to -\log 4$ as $N \to \infty$.

The proof for $\tilde{\theta}_d = \arg\max_{\theta_d} \int \mathcal{J}_d(\theta_d; \theta_g)\pi'(\theta_g|\tilde{\theta}_d, \mathcal{D})d\theta_g$ is similar to step 2 of the proof of Theorem 1. The corollary can then be concluded. □

S1.4 Adaptive Stochastic Gradient MCMC

Consider to solve the mean field equation:

$$h(\theta) = \int_{\mathcal{X}} H(\theta, \beta)\pi(\beta|\theta)d\beta = 0, \tag{S1.7}$$

where $\beta \in \mathcal{X}$ can be viewed a latent variable. Following Deng et al. (2019), we propose the following adaptive stochastic gradient MCMC algorithm for solving the equation (S1.7):

Algorithm 1 An adaptive stochastic gradient MCMC algorithm

1. $\beta_{k+1} = \beta_k + \epsilon_{k+1}(\nabla_\beta \tilde{L}(\beta_k, \theta_k) + \rho_k m_k) + \sqrt{2\epsilon\tau}\mathcal{N}(0, I)$,
2. $m_{k+1} = \alpha m_k + (1-\alpha)\nabla_\beta \tilde{L}(\beta_k, \theta_k)$,
3. $\theta_{k+1} = \theta_k + w_{k+1}H(\theta_k, \beta_{k+1})$,

In this algorithm, MSGLD (Kim et al., 2022) is used in drawing samples of β, $\nabla_\beta \tilde{L}(\beta_k, \theta_k)$ denotes an unbiased estimator of $\nabla_\beta \log \pi(\beta|\theta_k)$ obtained with the sample β_k, ϵ_{k+1} is called the learning rate used at iteration $k+1$, τ is the temperature, w_{k+1} is the step size used at iteration $k+1$, α is the momentum smoothing factor, and ρ_k is the momentum biasing factor. The algorithm is said "adaptive", as the parameter θ changes along with iterations.

Notations Algorithm 1 has the following notational correspondence with the EBGAN: (β, θ) in Algorithm 1 corresponds to (θ_g, θ_d) in the EBGAN; equation (S1.7) corresponds to

$$h(\theta_d) = \int H(\theta_d, \theta_g)\pi(\theta_g|\theta_d, \mathcal{D})d\theta_g = 0,$$

where $H(\theta_d; \theta_g)$ is as defined in (3.13), and $\pi(\theta_g|\theta_d, \mathcal{D}) \propto \exp(\mathbb{J}_g(\theta_g; \theta_d))q_g(\theta_g)$; $L(\beta, \theta)$ corresponds to $\log \pi(\theta_g|\theta_d, \mathcal{D})$ (up to an additive constant), and the stochastic gradient $\nabla_\beta \tilde{L}(\beta, \theta)$ in Algorithm 1 corresponds to $\nabla_{\theta_g} \tilde{L}(\theta_g, \theta_d)$ defined in (3.13).

S1.5 Convergence of the discriminator

To establish convergence of $\{\theta_k\}$ for Algorithm 1, we make the following assumptions.

Assumption S1. *(Conditions on stability and $\{\omega_k\}_{k\in\mathbb{N}}$) There exist a constant δ and a stationary point θ^* such that $\langle \theta - \theta^*, h(\theta)\rangle \leq -\delta\|\theta-\theta^*\|^2$ for any $\theta \in \Theta$. The step sizes $\{w_k\}_{k\in\mathbb{N}}$ form a positive decreasing sequence such that*

$$w_k \to 0, \quad \sum_{k=1}^{\infty} w_k = +\infty, \quad \liminf_{k\to\infty} 2\delta\frac{w_k}{w_{k+1}} + \frac{w_{k+1}-w_k}{w_{k+1}^2} > 0. \tag{S1.8}$$

Similar to Benveniste et al. (1990) (p.244), we can show that the following choice of $\{w_k\}$ satisfying (S1.8):

$$w_k = c_1/(c_2+k)^{\zeta_1}, \tag{S1.9}$$

for some constants $c_1 > 0$, $c_2 \geq 0$ and $\zeta_1 \in (0,1]$, provided that c_1 has been chosen large enough such that $2\delta c_1 > 1$ holds.

Assumption S2. *(Smoothness and Dissipativity) $L(\beta,\theta)$ is M-smooth on θ and β, and (m,b)-dissipative on β. In other words, for any $\beta, \beta_1, \beta_2 \in \mathcal{X}$*

and $\theta_1, \theta_2 \in \Theta$, the following inequalities hold:

$$\|\nabla_\beta L(\beta_1, \theta_1) - \nabla_\beta L(\beta_2, \theta_2)\| \leq M\|\beta_1 - \beta_2\| + M\|\theta_1 - \theta_2\|, \tag{S1.10}$$

$$\langle \nabla_\beta L(\beta, \theta), \beta \rangle \leq b - m\|\beta\|^2. \tag{S1.11}$$

Let β^* be a maximizer such that $\nabla_\beta L(\beta^*, \theta^*) = 0$, where θ^* is the stationary point defined in Assumption S1. By the dissipativity in Assumption S2, we have $\|\beta^*\|^2 \leq \frac{b}{m}$. Therefore,

$$\begin{aligned}\|\nabla_\beta L(\beta, \theta)\| &\leq \|\nabla_\beta L(\beta^*, \theta^*)\| + M\|\beta^* - \beta\| + M\|\theta - \theta^*\| \\ &\leq M\|\theta\| + M\|\beta\| + \bar{B},\end{aligned}$$

where $\bar{B} = M(\sqrt{\frac{b}{m}} + \|\theta^*\|)$. This further implies

$$\|L_\beta(\beta, \theta)\|^2 \leq 3M^2\|\beta\|^2 + 3M^2\|\theta\|^2 + 3\bar{B}^2. \tag{S1.12}$$

Assumption S3. *(Noisy gradient) Let $\xi_k = \nabla_\beta \tilde{L}(\beta_k, \theta_k) - \nabla_\beta L(\beta_k, \theta_k)$ denote the white noise contained in the stochastic gradient. The white noises $\xi_1, \xi_2, \ldots$ are mutually independent and satisfy the conditions:*

$$\begin{aligned}&E(\xi_k|\mathcal{F}_k) = 0, \\ &E\|\xi_k\|^2 \leq M^2 E\|\beta\|^2 + M^2 E\|\theta\|^2 + B^2,\end{aligned} \tag{S1.13}$$

where $\mathcal{F}_k = \sigma\{\theta_1, \beta_1, \theta_2, \beta_2, \ldots\}$ *denotes a* σ*-filter.*

The smoothness, dissipativity and noisy gradient conditions are regular for studying the convergence of stochastic gradient MCMC algorithms. Similar conditions have been used in many existing works such as Raginsky et al. (2017), Deng et al. (2019), and Gao et al. (2021).

Assumption S4. *(Boundedness) Assume that the trajectory of* θ *belongs to a compact set* Θ*, i.e.* $\{\theta_k\}_{k=1}^{\infty} \subset \Theta$ *and* $\|\theta_k\| \leq M$ *for some constant* M*.*

This assumption is more or less a technical condition. Otherwise, we can show that the Markov transition kernel used in Algorithm 1 satisfies the drift condition and, therefore, the varying truncation technique (see e.g. Chen and Zhu (1986); Andrieu et al. (2005)) can be employed in the algorithm for ensuring that $\{\theta_k : k = 1, 2, \ldots\}$ is almost surely contained in a compact space.

Lemma S1. *(Uniform* L_2 *bound) Suppose Assumptions S1-S4 hold. Given a sufficiently small learning rate* ϵ*, we have*

$$\sup_t E\|\beta_t\|^2 \leq G_\beta,$$

$$\sup_t E\langle \beta_t, m_t \rangle \leq G_m,$$

for some constants G_β *and* G_m*.*

Proof. We prove this lemma by mathematical induction under the weakest condition that both ϵ_t and ρ_t are set to constants. Assume that $E\|\beta_t\|^2 \leq G_\beta$ and $E\langle \beta_t, m_t\rangle \leq G_m$ for all $t = 1, \ldots, k$. By Algorithm 1, we have

$$
\begin{aligned}
&E\|\beta_{k+1}\|^2 = E\|\beta_k + \epsilon[\nabla_\beta \tilde{L}(\beta_k, \theta_k) + \rho m_k]\|^2 + 2\tau\epsilon d \\
&= E\|\beta_k + \epsilon[\nabla_\beta L(\beta_k, \theta_k) + \rho m_k]\|^2 + \epsilon^2 E\|\xi_k\|^2 + 2\tau\epsilon d \quad \text{(by Assumption S3)} \\
&= E\|\beta_k\|^2 + 2\epsilon E\langle \beta_k, \nabla_\beta L(\beta_k, \theta_k)\rangle + 2\rho\epsilon E\langle \beta_k, m_k\rangle + \epsilon^2 E\|\nabla_\beta L(\beta_k, \theta_k) + \rho m_k\|^2 \\
&\quad + \epsilon^2(M^2 E\|\beta_k\|^2 + M^2 E\|\theta_k\|^2 + B^2) + 2\tau\epsilon d,
\end{aligned}
\tag{S1.14}
$$

where d is the dimension of β. Further, we can show that $m_k = (1-\alpha)\nabla_\beta \tilde{L}(\beta_{k-1}, \theta_{k-1}) + \alpha(\alpha-1)\nabla_\beta \tilde{L}(\beta_{k-2}, \theta_{k-2}) + \alpha^2(\alpha-1)\nabla_\beta \tilde{L}(\beta_{k-3}, \theta_{k-3}) + \cdots$. By Assumption S2-S3 and equation (S1.12), for any $i \geq 1$, we have $E\|\nabla_\beta \tilde{L}(\beta_{k-i}, \theta_{k-i})\|^2 \leq E\|\nabla_\beta L(\beta_{k-i}, \theta_{k-i})\|^2 + E\|\xi_{k-i}\|^2 \leq 4M^2 E\|\beta_{k-i}\|^2 + 4M^2 E\|\theta\|^2 + 3\bar{B}^2 + B^2 \leq 4M^2 G_\beta + 4M^4 + 3\bar{B}^2 + B^2$. Therefore,

$$
\begin{aligned}
&E\|m_k\|^2 = \sum_{i=1}^{k}[(1-\alpha)\alpha^{i-1}]^2 E\|\nabla_\beta \tilde{L}(\beta_{k-i}, \theta_{k-i})\|^2 \\
&+ 2\sum_{1\leq i,j\leq k}[(1-\alpha)\alpha^{i-1}][(1-\alpha)\alpha^{j-1}]\sqrt{E\|\nabla_\beta \tilde{L}(\beta_{k-i}, \theta_{k-i})\|^2}\sqrt{E\|\nabla_\beta \tilde{L}(\beta_{k-j}, \theta_{k-j})\|^2} \\
&\leq 4M^2 G_\beta + 4M^4 + 3\bar{B}^2 + B^2.
\end{aligned}
\tag{S1.15}
$$

Combined with (S1.14), this further implies

$$\begin{aligned}
E\|\beta_{k+1}\|^2 \leq & E\|\beta_k\|^2 + 2\epsilon E(b - m\|\beta_k\|^2) + 2\rho\epsilon G_m \\
& + 2\epsilon^2(3M^2E\|\beta_k\|^2 + 3M^4 + 3\bar{B}^2) + 2\epsilon^2\rho^2(4M^2G_\beta + 4M^4 + 3\bar{B}^2 + B^2) \\
& + \epsilon^2(M^2E\|\beta_k\|^2 + M^2E\|\theta_k\|^2 + B^2) + 2\tau\epsilon d \\
= & (1 - 2\epsilon m + 7M^2\epsilon^2)E\|\beta_k\|^2 + 2\epsilon b + 2\rho\epsilon G_m + 2\tau\epsilon d + 2\epsilon^2(3M^4 + 3\bar{B}^2) \\
& + 2\epsilon^2\rho^2(4M^2G_\beta + 4M^4 + 3\bar{B}^2 + B^2) + \epsilon^2(M^4 + B^2).
\end{aligned} \tag{S1.16}$$

On the other hand,

$$\begin{aligned}
& E\langle\beta_{k+1}, m_{k+1}\rangle = E\langle\beta_k + \epsilon[\nabla_\beta\tilde{L}(\beta_k, \theta_k) + \rho m_k], \alpha m_k + (1-\alpha)\nabla_\beta\tilde{L}(\beta_k, \theta_k)\rangle \\
\leq & \alpha E\langle\beta_k, m_k\rangle + E\langle\beta_k, (1-\alpha)\nabla_\beta L(\beta_k, \theta_k)\rangle + \epsilon(1+\rho)\max\{E\|\nabla_\beta\tilde{L}(\beta_k, \theta_k)\|^2, E\|m_k\|^2\} \\
\leq & \alpha G_m + (1-\alpha)b + \epsilon(1+\rho)(4M^2G_\beta + 4M^4 + 3\bar{B}^2 + B^2).
\end{aligned} \tag{S1.17}$$

To induce mathematical induction, following from (S1.16) and (S1.17), it is sufficient to show

$$\begin{aligned}
G_\beta \leq & \frac{1}{2\epsilon m - 7M^2\epsilon^2 - 8\epsilon^2\rho^2M^2}\Big\{2\epsilon b + 2\rho\epsilon G_m + 2\tau\epsilon d \\
& + 2\epsilon^2(3M^4 + 3\bar{B}^2) + 2\epsilon^2\rho^2(4M^4 + 3\bar{B}^2 + B^2) + \epsilon^2(M^4 + B^2)\Big\}, \\
G_m \leq & \frac{1}{1-\alpha}\Big\{(1-\alpha)b + \epsilon(1+\rho)(4M^2G_\beta + 4M^4 + 3\bar{B}^2 + B^2)\Big\}.
\end{aligned}$$

When ϵ is sufficiently small, it is not difficult to see that the above inequalities holds for some G_β and G_m. □

Assumption S5. *(Lipschitz condition of $H(\theta, \beta)$) $H(\theta, \beta)$ is Lipschitz continuous on β; i.e., there exists a constant M such that*

$$\|H(\theta, \beta_1) - H(\theta, \beta_2)\| \leq M\|\beta_1 - \beta_2\|.$$

By Assumption S5, $\|H(\theta_k, \beta_{k+1})\|^2 \leq 2M\|\beta_{k+1}\|^2 + 2\|H(\theta_k, 0)\|^2$. Since θ_k belongs to a compact set and $H(\theta, 0)$ is a continuous function, there exists a constant B such that

$$\|H(\theta_k, \beta_{k+1})\|^2 \leq 2M^2\|\beta_{k+1}\|^2 + 2B^2. \tag{S1.18}$$

Assumption S6. *(Solution of Poisson equation) For any $\theta \in \Theta$, $\beta \in \mathcal{X}$, and a function $V(\beta) = 1 + \|\beta\|$, there exists a function $\mu_\theta(\beta)$ that solves the Poisson equation $\mu_\theta(\beta) - \mathcal{T}_\theta\mu_\theta(\beta) = H(\theta, \beta) - h(\theta)$ such that*

$$H(\theta_k, \beta_{k+1}) = h(\theta_k) + \mu_{\theta_k}(\beta_{k+1}) - \mathcal{T}_{\theta_k}\mu_{\theta_k}(\beta_{k+1}), \quad k = 1, 2, \ldots, \tag{S1.19}$$

where $\mathcal{T}_\theta$ is the probability transition kernel and $\mathcal{T}_\theta\mu_\theta(\beta) = \int \mu_\theta(\beta')\mathcal{T}_\theta(\beta, d\beta')$. Moreover, for all $\theta, \theta' \in \Theta$ and $\beta \in \mathcal{X}$, we have $\|\mu_\theta(\beta) - \mu_{\theta'}(\beta)\| \leq \varsigma_1\|\theta - \theta'\|V(\beta)$ and $\|\mu_\theta(\beta)\| \leq \varsigma_2 V(\beta)$ for some constants $\varsigma_1 > 0$ and $\varsigma_2 > 0$.

This assumption has often been used in the study for the convergence of the SGLD algorithm, see e.g. Whye et al. (2016) and Deng et al. (2019). Alternatively, as mentioned above, we can show that the Markov transition kernel used in Algorithm 1 satisfies the drift condition and thus Assumption S6 can be verified as in Andrieu et al. (2005).

Proof of Lemma 2

Proof. Our proof follows the proof of Theorem 1 in Deng et al. (2019). However, Algorithm 1 employs MSGLD for updating β, while Deng et al. (2019) employs SGLD. We replace Lemma 1 of Deng et al. (2019) by Lemma S1 to accommodate this difference. In addition, Proposition 3 and Proposition 4 in Deng et al. (2019) are replaced by equation (S1.12) and equation (S1.18) respectively.

Further, based on the proof of Deng et al. (2019), we can derive an explicit formula for γ:

$$\gamma = \gamma_0 + 12\sqrt{3}M\left((2M^2 + \varsigma_2^2)G_\beta + 2B^2 + \varsigma_2^2\right)^{\frac{1}{2}}, \tag{S1.20}$$

where γ_0 together with t_0 can be derived from Lemma 3 of Deng et al. (2019) and they depend on δ and $\{\omega_t\}$ only. The second term of γ is obtained by applying the Cauchy-Schwarz inequality to bound the expectation $E\langle\theta_t - \theta^*, \mathcal{T}_{\theta_{t-1}}\mu_{\theta_{t-1}}(\beta_t)\rangle$, where $E\|\theta_t - \theta^*\|^2$ can be bounded by $2M^2$ by Assumption S4 and $E\|\mathcal{T}_{\theta_{t-1}}\mu_{\theta_{t-1}}(\beta_t)\|^2$ can be bounded according to (S1.19) and the upper bound of $H(\theta, \beta)$ given in (S1.18). □

S1.6 Convergence of the Generator

To establish the weak convergence of β_t in Algorithm 1, we need more assumptions. Let the fluctuation between ψ and $\bar{\psi}$:

$$\mathcal{L}f(\theta) = \psi(\theta) - \bar{\psi}, \tag{S1.21}$$

where $f(\theta)$ is the solution to the Poisson equation, and $\mathcal{L}$ is the infinitesimal generator of the Langevin diffusion

Assumption S7. *Given a sufficiently smooth function $f(\theta)$ as defined in (S1.21) and a function $\mathcal{V}(\theta)$ such that the derivatives satisfy the inequality $\|D^j f\| \lesssim \mathcal{V}^{p_j}(\theta)$ for some constant $p_j > 0$, where $j \in \{0, 1, 2, 3\}$. In addition, $\mathcal{V}^p$ has a bounded expectation, i.e., $\sup_k E[\mathcal{V}^p(\theta_k)] < \infty$; and $\mathcal{V}^p$ is smooth, i.e. $\sup_{s\in(0,1)} \mathcal{V}^p(s\theta + (1-s)\vartheta) \lesssim \mathcal{V}^p(\theta) + \mathcal{V}^p(\vartheta)$ for all $\theta, \vartheta \in \Theta$ and $p \leq 2\max_j\{p_j\}$.*

Proof of Lemma 3

Proof. The update of β can be rewritten as

$$\beta_{k+1} = \beta_k + \epsilon_{k+1}(\nabla_\beta L(\beta_k, \tilde{\theta}_d) + \Delta\tilde{V}_k) + \sqrt{2\epsilon\tau}\mathcal{N}(0, I),$$

where $\Delta\tilde{V}_k = \nabla_\beta L(\beta_k, \theta_k) - \nabla_\beta L(\beta_k, \tilde{\theta}_d) + \xi_k + \rho_k m_k$ can be viewed as the

estimation error of $\nabla_\beta \tilde{L}(\beta_k, \theta_k)$ for the "true" gradient $\nabla_\beta L(\beta_k, \tilde{\theta}_d)$. For the terms in $\Delta\tilde{V}_k$, by Lemma 2 and Assumption S2, we have

$$\mathbb{E}\|\nabla_\beta L(\beta_k, \theta_k) - \nabla_\beta L(\beta_k, \tilde{\theta}_d)\| \leq M\mathbb{E}\|\theta_k - \tilde{\theta}_d\| \leq M\sqrt{\gamma\omega_k} \to 0;$$

by Assumption 3 and Lemma S1, $\mathbb{E}\|\xi_k\|^2 \leq M^2\mathbb{E}\|\beta\|^2 + M^2\mathbb{E}\|\theta\|^2 + B^2$ is upper bounded; and as implied by (S1.15), there exists a constant C such that $\mathbb{E}\|\rho_k m_k\| \leq C\rho_k$. Then parts (i) and (ii) can be concluded by applying Theorem 5 and Theorem 3 of Chen et al. (2015), respectively, where the proofs only need to be slightly modified to accommodate the convergence of $\theta_k \to \tilde{\theta}_d$ (as shown in Lemma 2) and the momentum biasing factor ρ_k. □

S2 Evaluation Metrics for Generative Adversarial Networks

The inception scores (IS) (Salimans et al., 2016), Wasserstein distance (WD), and maximum mean discrepancy (MMD) are metrics that are often used for assessing the quality of images generated by a generative image model. See Xu et al. (2018) for an empirical evaluation on them.

Let $p_g(x)$ be a probability distribution of the images generated by the model, and let $p_{dis}(y|x)$ be the probability that image x has label y according

to a pretrained discriminator. The IS of p_{gen} relative to p_{dis} is given by

$$IS(p_g) = \exp\left\{\mathbb{E}_{x\sim p_g} D_{KL}(p_{dis}(y|x)|\int p_{dis}(y|x)p_g(x)dx\right\},$$

which takes values in the interval $[1, m]$ with m being the total number of possible labels. A higher IS value is preferred as it means p_g is a sharp and distinct collection of images. To calculate IS, we employed transfer learning to obtain a pretrained discriminator, which involves retraining the pretrained ResNet50, the baseline model, on Fashion MNIST data by tuning the weights on the first and last hidden layers.

The first moment Wasserstein distance, denoted by 1-WD in the paper, for the two distributions p_g and p_{data} is defined as

$$WD(p_g, p_{data}) = \inf_{\gamma\in\Gamma(p_g,p_{data})} \mathbb{E}_{x_g\sim p_g, x_r\sim p_{data}} \|x_g - x_r\|,$$

where $\Gamma(p_g, p_{data})$ denotes the set of all joint distributions with the respective marginals p_g and p_{data}. The 1-WD also refers to the earth mover's distance. Let $\{x_{g,i} : i = 1, 2, \ldots, n\}$ denote n samples drawn from p_g, and let $\{x_{r,i} : i = 1, 2, \ldots, n\}$ denote n samples drawn from p_{data}. With the samples, the 1-WD can be calculated by solving the optimal transport

problem:

$$WD(p_g, p_{data}) = \min_{w \in \mathbb{R}^{n \times n}} \sum_{i=1}^{n} \sum_{j=1}^{n} w_{ij} \|x_{g,i} - x_{r,j}\|,$$
$$s.t. \quad \sum_{j=1}^{n} w_{ij} = p_g(x_{g,i}), \forall i; \quad \sum_{i=1}^{n} w_{ij} = p_{data}(x_{r,j}), \forall j.$$

To calculate Wasserstein distance, we used the code provided at `https://github.com/xuqiantong/GAN-Metrics/`.

To address the computational complexity of 1-WD, which is of $O(n^3)$, we partitioned the samples drawn at each run to 1000 groups, each group being of size 100, and calculated 1-WD for each group and then average the distance over the groups. The distance values from each run were further averaged over five independent runs and reported in Table 1 of the main text.

The MMD measures the dissimilarity between the two distributions p_g and p_{data} for some fixed kernel function $\kappa(\cdot, \cdot)$, and it is defined as

$$MMD^2(p_g, p_{data}) = \mathbb{E}_{x_g, x'_g \sim p_g; x_r, x'_r \sim p_{data}}[\kappa(x_g, x'_g) - 2\kappa(x_g, x_r) + \kappa(x_r, x'_r)].$$

A lower MMD value means that p_g is closer to p_{data}. In this paper, we calculated MMD values using the code provided at `https://www.onurtunali.com/ml/2019/03/08/maximum-mean-discrepancy-in-machine-learning`.

`html` with the "rbf" kernel option. We calculated the MMD values with the same sample grouping method as used in calculation of 1-WD.

S3 More Numerical Examples

S3.1 A Gaussian Example: Additional Results

Figure S1 shows the empirical means of $D_{\theta_d^{(t)}}(x)$ and $D_{\theta_d^{(t)}}(\tilde{x})$ produced by the two methods along with iterations, which indicates that both methods can reach the 0.5-0.5 convergence very fast.

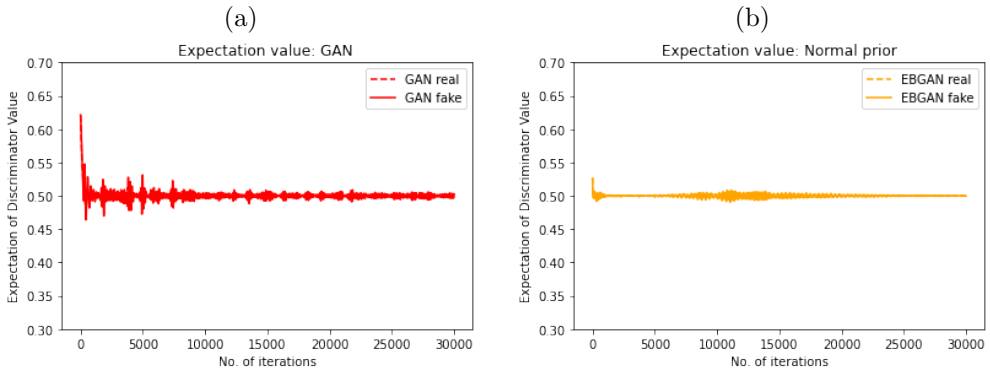


Figure S1: Empirical means of $D_{\theta_d^{(t)}}(x_i)$ and $D_{\theta_d^{(t)}}(\tilde{x}_i)$ produced by (a) GAN and (b) EBGAN with a Gaussian prior along with iterations.

S3.2 A Mixture Gaussian Example: Additional Results

For this example, we have tried the choice $\phi_3(D) = \log(D)$ of non-saturating GAN. Under this non-saturating setting, the game is no longer of the

minimax style. However, it helps to overcome the gradient vanishing issue suffered by the minimax GAN. Figure S2 shows the empirical means $\mathbb{E}(D_{\theta_d^{(t)}}(x_i))$ and $\mathbb{E}(D_{\theta_d^{(t)}}(\tilde{x}_i))$ produced by different methods along with iterations. The non-saturating GAN and Lipschitz GAN still failed to converge to the Nash equilibrium, but BGAN and ProbGAN nearly converged after about 2000 iterations. In contrast, EBGAN still worked very well: It can converge to the Nash equilibrium in either case, with or without a Lipschitz penalty.

Figure S3 shows the plots of component recovery from the fake data. It indicates that EBGAN has recovered all 10 components of the real data in either case, with or without a Lipschitz penalty. Both ProbGAN and BGAN worked much better with this non-saturating choice than with the minimax choice of ϕ_3: The ProbGAN has even recovered all 10 components, although the coverage area is smaller than that by EBGAN; and BGAN just had one component missed in recovery. The non-saturating GAN and Lipschitz GAN still failed for this example, which is perhaps due to the the model collapse issue. Using a single generator is hard to generate data following a multi-modal distribution.

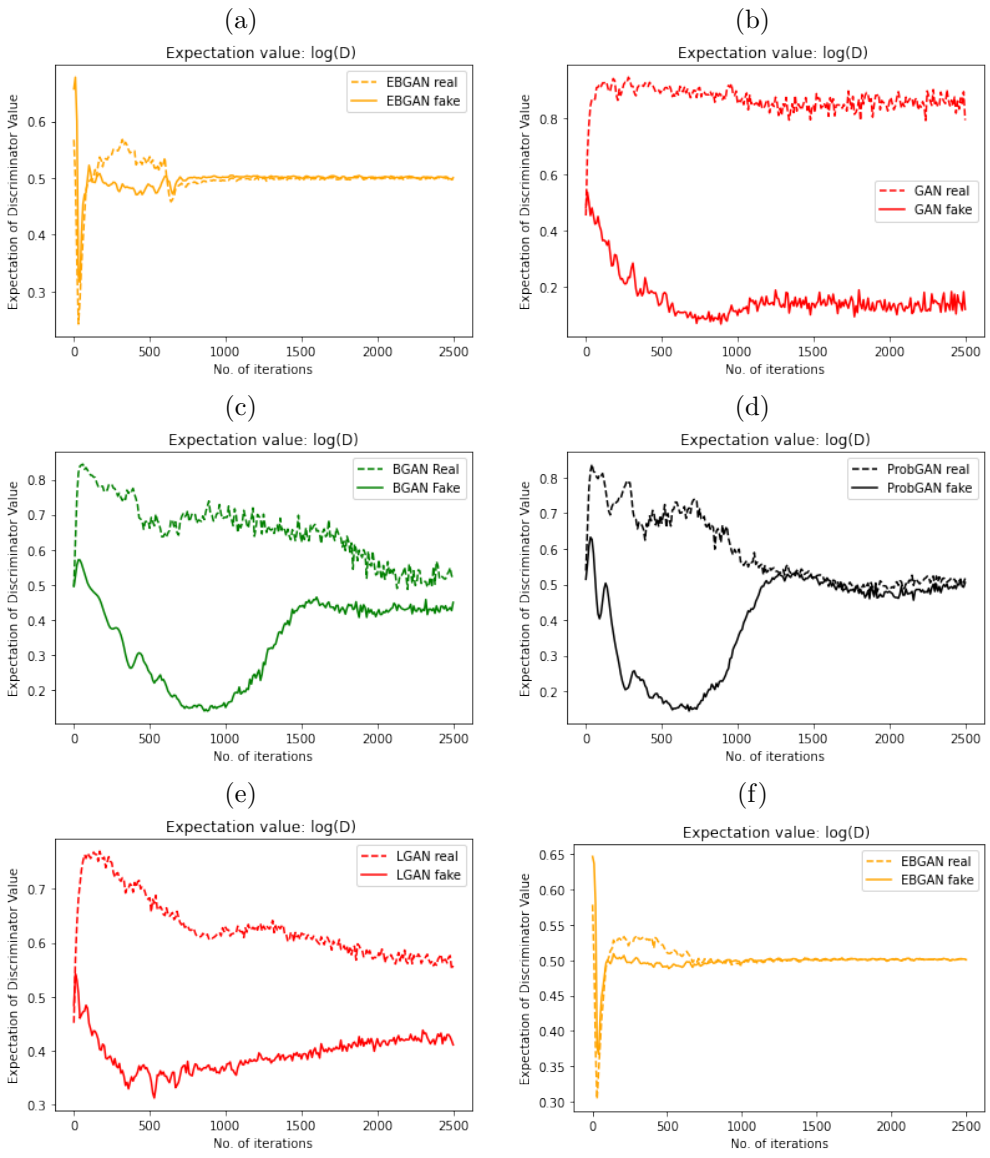


Figure S2: Nash equilibrium convergence plots with $\phi_3(D) = \log(D)$, which compare the empirical means of $D_{\theta_d^{(t)}}(x_i)$ and $D_{\theta_d^{(t)}}(\tilde{x}_i)$ produced by different methods along with iterations: (a) EBGAN with $\lambda = 0$, (b) non-saturating GAN, (c) BGAN, (d) ProbGAN, (e) Lipschitz GAN, and (f) EBGAN with a Lipschitz penalty.

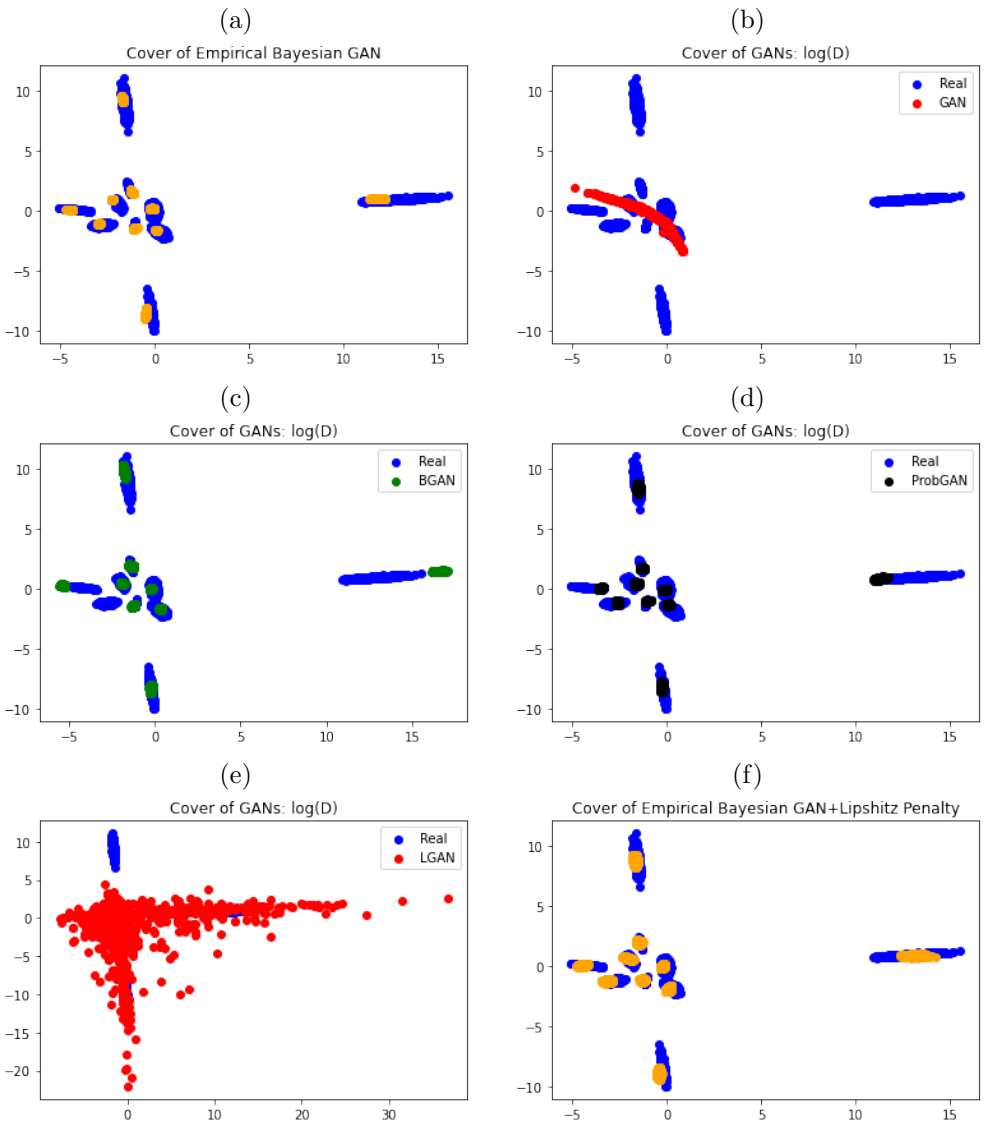


Figure S3: Component recovery plots produced by different methods with $\phi_3(D) = \log(D)$: (a) EBGAN with $\lambda = 0$, (b) non-saturating GAN, (c) BGAN, (d) ProbGAN, (e) Lipschitz GAN, and (f) EBGAN with a Lipschitz penalty.

S3.3 Image Generation: HMNIST

We compared GAN and EBGAN on another real data problem, the HAM10000 ("Human Against Machine with 10000 training images") dataset, which is also known as HMNIST and available at `https://www.kaggle.com/kmader/skin-cancer-mnist-ham10000`. The dataset consists of a total of 10,015 dermatoscopic images of skin lesions classified to seven types of skin cancer. Unlike other benchmark computer vision datasets, HMNIST has imbalanced group sizes. The largest group size is 6705, while the smallest one is 115, which makes it hard for conventional GAN training algorithms.

Our results for the example are shown in Figure S4, which indicates again that the EBGAN outperforms the GAN. In particular, the GAN is far from the 0.5-0.5 convergence, while EBGAN can achieve it. In terms of images generated by the two methods, it is clear that GAN suffers from a mode collapse issue; many images generated by it have a similar pattern even, e.g., those shown in the cells (1,4), (2,1), (4,1), (4,5), (5,2), (5,5) and (6,2) share a similar pattern. In contrast, the images generated by EBGAN show a clear clustering structure; each row corresponds to one different pattern.

Since the clusters in the dataset are imbalanced, the IS score does not work well for measuring the quality of the generated images. To tackle

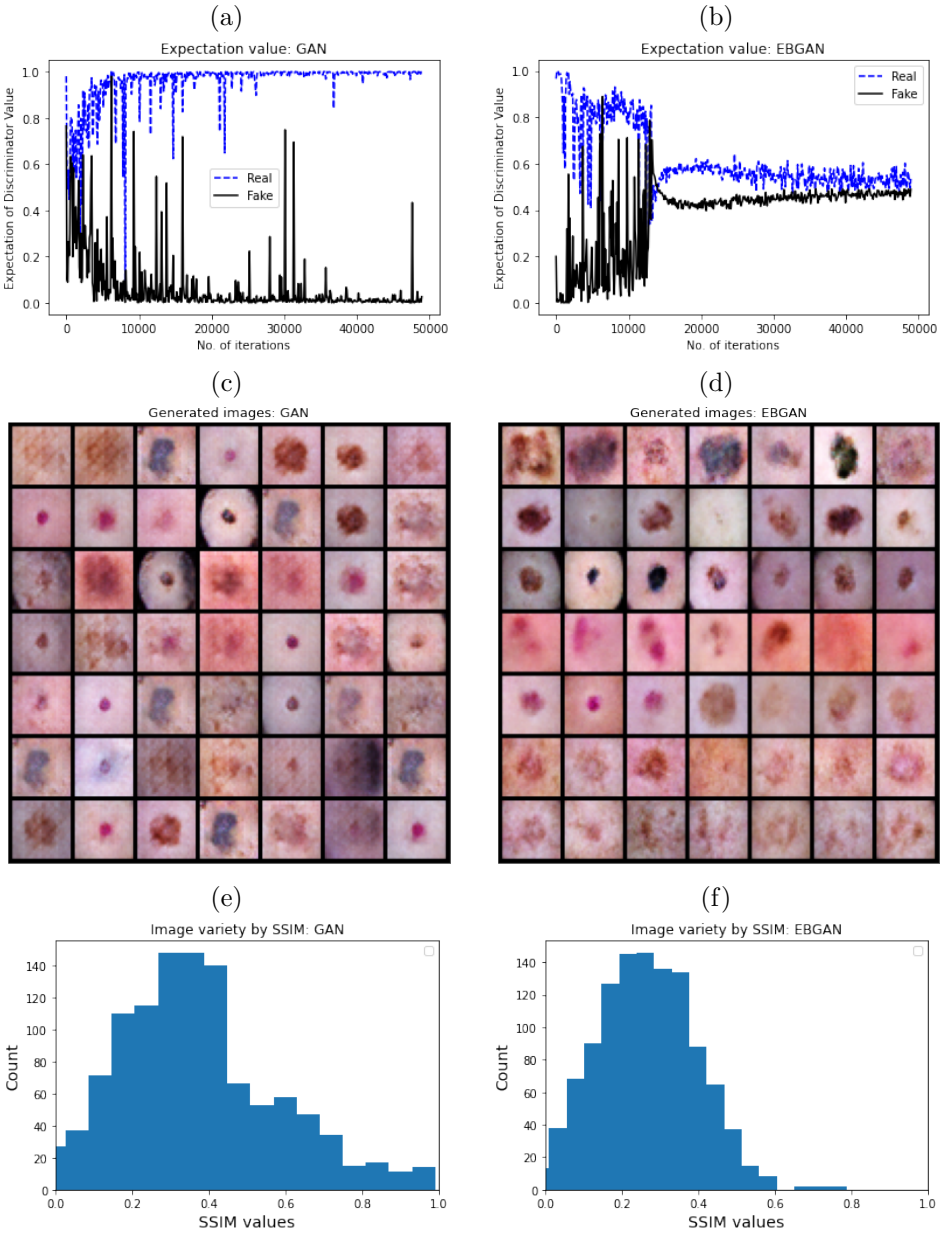


Figure S4: Results for the HMNIST example: (a) convergence plot of GAN; (b) convergence plot of EBGAN; (c) images generated by GAN; (d) images generated by EBGAN; (e) histograms of SSIMs for the images shown in plot (c) ; and (f) histograms of SSIMs for the images shown in plot (d).

this issue, we calculated the structural similarity index measure (SSIM) Wang et al. (2004) for each pair of the images shown in Figure S4(c) and those shown in Figure S4(d), respectively. SSIM is a metric that measures the similarity between two images; it takes a value of 1 if two images are identical. Figure S4(e) & (f) shows the histograms of SSIMs for the images shown in Figure S4(c) & (d), respectively. The comparison shows clearly that the images generated by EBGAN have a larger diversity than those by GAN.

S3.4 Conditional Independence Tests

Conditional independence is a fundamental concept in graphical modeling (Lauritzen, 1996) and causal inference (Pearl, 2009) for multivariate data. Conditional independence tests have long been studied in statistics, which are to test the hypotheses

$$H_0 : \ X \perp\!\!\!\perp Y|Z \quad versus \quad H_1 : \ X \not\perp\!\!\!\perp Y|Z,$$

where $X \in \mathbb{R}^{d_x}$, $Y \in \mathbb{R}^{d_y}$ and $Z \in \mathbb{R}^{d_z}$. For the case that the variables are discrete and the dimensions are low, the Pearson χ^2-test and the likelihood ratio tests are often used. For the case that the variables are Gaussian and linearly dependent, one often conducts the test using the partial correlation

coefficient or its equivalent measure, see e.g., Spirtes et al. (1993) and Liang et al. (2015). However, in real-life situations, the normality and linear dependence assumptions are often not satisfied and thus nonparametric conditional independence tests are required. An abundance of such type of tests have been developed in the literature, e.g., permutation-based tests (Doran et al., 2014; Berrett et al., 2019), kernel-based tests (Zhang et al., 2012; Strobl et al., 2019), classification or regression-based tests (Sen et al., 2017; Zhang et al., 2017), and knockoff tests (Candès et al., 2018). Refer to Li and Fan (2019) for an overview.

As pointed out in Li and Fan (2019), the existing nonparametric conditional independence tests often suffer from the curse of dimensionality in the confounding vector Z; that is, the tests may be ineffective when the sample size is small, since the accumulation of spurious correlations from a large number of variables in Z makes it difficult to discriminate between the hypotheses. As a remedy to this issue, Bellot and van der Schaar (2019) proposed a generative conditional independent test (GCIT) based on GAN. The method belongs to the class of nonparametric conditional independence tests and it consists of three steps: (i) simulating samples $\tilde{X}_1, \ldots, \tilde{X}_M \sim q_{H_0}(X)$ under the null hypothesis H_0 via GAN, where $q_{H_0}(X)$ denotes the distribution of X under H_0; (ii) defining an ap-

propriate test statistic $\varrho(\cdot)$ which captures the X-Y dependency in each of the samples $\{(\tilde{X}_1, Y, Z), (\tilde{X}_2, Y, Z), \ldots, (\tilde{X}_M, Y, Z)\}$; and (iii) calculating the p-value

$$\widehat{p} = \frac{\sum_{m=1}^{M} \mathbf{1}\left\{\varrho(\tilde{\mathbf{X}}_{\mathbf{m}}, \mathbf{Y}, \mathbf{Z}) > \varrho(\mathbf{X}, \mathbf{Y}, \mathbf{Z})\right\}}{M}, \tag{S3.22}$$

which can be made arbitrarily close to the true probability

$$\mathbb{E}_{\tilde{X} \sim q_{H_0}(X)} \mathbf{1}\left\{\varrho(\tilde{\mathbf{X}}, \mathbf{Y}, \mathbf{Z}) \geq \varrho(\mathbf{X}, \mathbf{Y}, \mathbf{Z})\right\}$$

by sampling a large number of samples $\tilde{X}$ from $q_{H_0}(X)$. Bellot and van der Schaar (2019) proved that this test is valid and showed empirically that it is robust with respect to the dimension of the confounding vector Z. It is obvious that the power of the GCIT depends on how well the samples $\{\tilde{X}_1, \tilde{X}_2, \ldots, \tilde{X}_m\}$ approximate the distribution $q_{H_0}(X)$.

Simulation Studies

To show that EBGAN improves the testing power of GCIT, we consider a simulation example taken from Bellot and van der Schaar (2019) for testing the hypotheses:

$$H_0: \quad X = f_1(A_x Z + \epsilon_x), \quad Y = f_2(A_y Z + \epsilon_y),$$

$$H_1: \quad X = f_1(A_x Z + \epsilon_x), \quad Y = f_3(\alpha A_{xy} X + A_y Z + \epsilon_y),$$

where the matrix dimensions of $A_{(\cdot)}$ are such that X and Y are univariate. The entries of $A_{(\cdot)}$ as well as the parameter α are randomly drawn from Unif$[0, 1]$, and the noise variables $\epsilon_{(\cdot)}$ are Gaussian with mean 0 and variance 0.025. Three specific cases are considered in the simulation:

- Case 1. Multivariate Gaussian: f_1, f_2 and f_3 are identity functions, $Z \sim \mathcal{N}(0, I_{d_z})$, which result in multivariate Gaussian data and linear dependence under H_1.

- Case 2. Arbitrary relationship: f_1, f_2 and f_3 are randomly sampled from $\{\tanh(x), \exp(-x), x^2\}$, $Z \sim N(0, I_{d_z})$, which results in more complex distributions and variable dependencies. It resembles the complexities we can expect in real applications.

- Case 3. Arbitrary relationships with a mixture Z distribution:

  $$H_0: \quad X = \{f_{1,a}(A_x Z_a + \epsilon_x), f_{1,b}(A_x Z_b + \epsilon_x)\}, \quad Y = f_2(A_y Z + \epsilon_y),$$
  $$H_1: \quad X = \{f_{1,a}(A_x Z_a + \epsilon_x), f_{1,b}(A_x Z_b + \epsilon_x)\}, \quad Y = f_3(\alpha A_{xy} X + A_y Z + \epsilon_y),$$

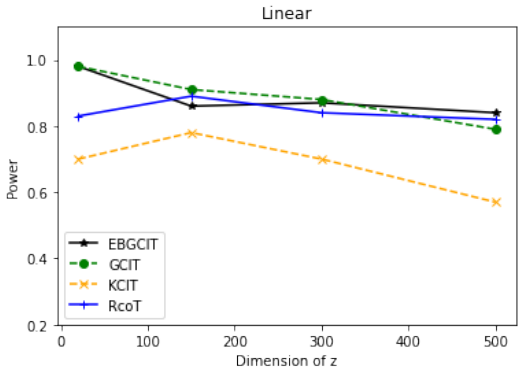
  where $Z_a \sim \mathcal{N}(1_d, I_d)$, $Z_b \sim \mathcal{N}(-1_d, I_d)$, $Z_a, Z_b \in \mathbb{R}^{\frac{n}{2} \times d}$, $Z = (Z_a^T, Z_b^T)^T$, $f_{1,a}$, $f_{1,b}$, f_2 and f_3 are randomly sampled from $\{\tanh(x), \exp(-x), x^2\}$ and $f_{1,a} \neq f_{1,b}$.

For each case, we simulated 100 datasets under H_1, where each dataset

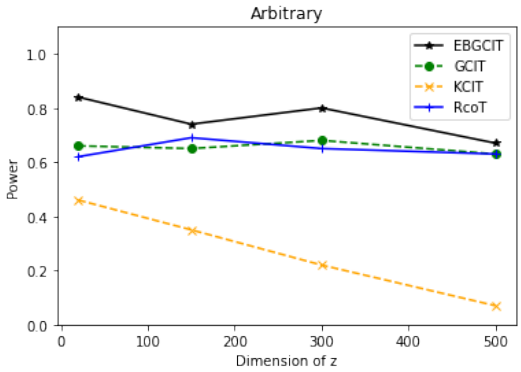
consisted of 150 samples. Both GAN and EBGAN were applied to this example with the randomized dependence coefficient (Lopez-Paz et al., 2013) used as the test statistic $\varrho(\cdot)$. Here GAN was trained as in Bellot and van der Schaar (2019) with the code available at `https://github.com/alexisbellot/GCIT`. In GAN, the objective function of the generator was regularized by a mutual information which encourages to generate samples $\tilde{X}$ as independent as possible from the observed variables X and thus enhances the power of the test. The EBGAN was trained in a plain manner without the mutual information term included in $\mathcal{J}(\theta_d; \theta_g)$. Detailed settings of the experiments were given in the supplement. In addition, two kernel-based methods, KCIT (Zhang et al., 2012) and RCoT (Strobl et al., 2019), were applied to this example for comparison.

Figure S5 summarizes the results of the experiments. For case 1, EBGAN, GAN and RCoT are almost the same, and they all outperform KCIT. For case 2 and case 3, EBGAN outperforms the other three methods significantly. Note that, by Bellot and van der Schaar (2019), GAN represents the state-of-the-art method for high-dimensional nonparametric conditional independence tests. For similar examples, Bellot and van der Schaar (2019) showed that GAN significantly outperformed the existing statistical tests, including the kernel-based tests (Zhang et al., 2012; Strobl

(a)



(b)



(c)

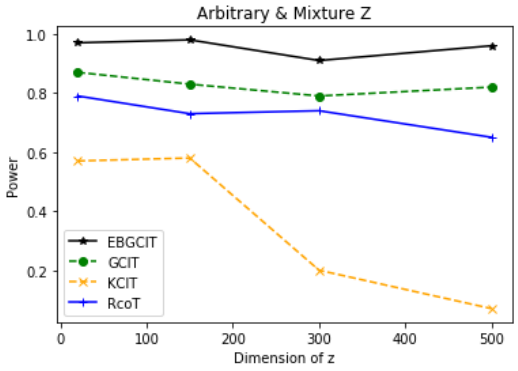


Figure S5: Generative conditional independence tests with GAN (denoted by GCIT) and EBGAN (denoted by EBGCIT): (a) Power curve for Case 1; (b) Power curve for Case 2 ; (c) Power curve for Case 3

et al., 2019), knockoff-based test (Candès et al., 2018), and classification-based test (Sen et al., 2017).

Identifications of Drug Sensitive Mutations

As a real data example, we applied EBGAN to identification of genetic mutations that affect response of cancer cells to an anti-cancer drug. This helps cancer clinics, as the treatment for a cancer patient can strongly depend on the mutation of his/her genome (Garnett et al., 2012) in precision medicine. We used a sub-dataset of Cancer Cell Line Encyclopedia (CCLE), which relates the drug response of PLX4720 with 466 genetic mutations. The dataset consists of 474 cell lines. Detailed settings of the experiment were given in the supplement.

Table S1 shows the mutations identified by EBGAN at a significance level of 0.05, where the dependency of the drug response on the first 12 mutations has been validated by the existing literature at PubMed. Since PLX4720 was designed as a BRAF inhibitor, the low p-values of BRAF.MC, BRAF.V600E and BRAF confirm the validity of the proposed test. EBGAN also identified MYC as a drug sensitive mutation, but which was not detected via GAN in Bellot and van der Schaar (2019). Our finding is validated by Singleton et al. (2017), which reported that BRAF mutant cell

lines with intrinsic resistance to BRAF rapidly upregulate MYC upon treatment of PLX4720. CRKL is another mutation identified by EBGAN but not by GAN, and this finding can be validated by the experimental results reported in Tripathi et al. (2020).

Table S1: Genetic experiment results: Each cell gives the p-value indicating the dependency between a mutation and drug response, where the superscript $^-$ indicates that the dependency of drug response on the mutation has not yet been validated in the literature.

BRAF.MC	IRAK1	BRAF.V600E	BRAF	HIP1	SRPK3	MAP2K4	FGR
0.001	0.002	0.003	0.003	0.004	0.012	0.014	0.014
PRKD1	CRKL	MPL	MYC	MTCP1$^-$	ADCK2$^-$	RAD51L1$^-$	
0.015	0.016	0.027	0.037	0.011	0.037	0.044	

S3.5 Nonparametric Clustering

This section gives details for different datasets we tried.

Two-Circle Problem

The most notorious example for classical clustering methods is the two-circle problem. The dataset is generated as follows:

$$
\begin{aligned}
&Z_i = (z_{1i}, z_{2i}), \quad \text{where} \quad \mathrm{z_{1i}, z_{2i}} \overset{iid}{\sim} \mathrm{Unif}[-1, 1], \quad \mathrm{i} = 1, \ldots, 1000; \\
&\text{Inner Circle} : 0.25 * \left(\frac{z_{1i}}{\sqrt{z_{1i}^2 + z_{2i}^2}}, \frac{z_{2i}}{\sqrt{z_{1i}^2 + z_{2i}^2}} \right) + \epsilon, \quad i = 1, \ldots, 500; \\
&\text{Outer Circle} : \left(\frac{z_{1i}}{\sqrt{z_{1i}^2 + z_{2i}^2}}, \frac{z_{2i}}{\sqrt{z_{1i}^2 + z_{2i}^2}} \right) + \epsilon, \quad i = 501, \ldots, 1000,
\end{aligned}
\tag{S3.23}
$$

where $\epsilon \sim \mathcal{N}(0, 0.05^2 I_2)$. For this example, the K-means and agglomerative clustering methods are known to fail to detect the inner circle unless the data are appropriately transformed; DBSCAN is able to detect the inner circle, but it is hard to apply to other high-dimensional problems due to its density estimation-based nature.

For a simulated dataset, each of the methods, including K-means, agglomerative, DBSCAN, Cluster GAN and Cluster EBGAN, was run for 100 times with different initializations. Figure S6 shows the histogram of the adjust Rand index (ARI) (Rand, 1971) values obtained in those runs. It indicates that K-means, agglomerative and DBSCAN produced the same clustering results in different runs, while Cluster GAN and Cluster EBGAN

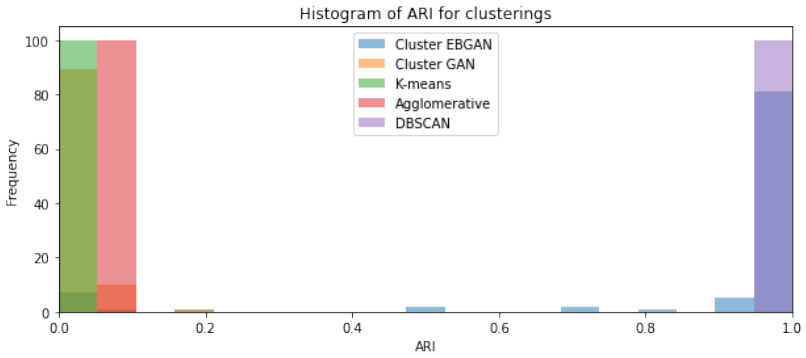


Figure S6: Histogram of ARI produced by different methods: Cluster EBGAN, Cluster GAN, K-means, Agglomerative, and DBSCAN.

produced different ones in different runs. In particular, the ARI values resulted from Cluster GAN are around 0, whereas those from Cluster EBGAN are around 1.0. Figure S7 shows some clustering results produced by these methods.

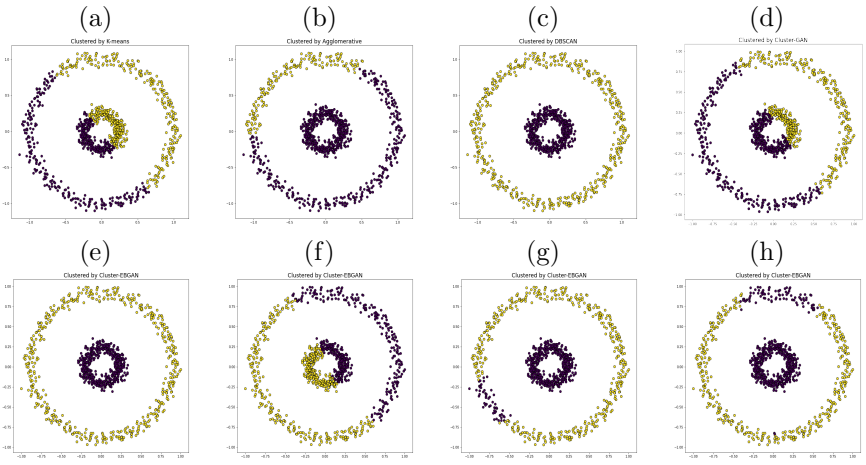


Figure S7: (a) K-means clustering, (b) Agglomerative clustering, (c) DBSCAN, (d) Cluster-GAN, (e)-(h) Cluster-EBGAN in different runs.

Figure S6 and Figure S7 indicate that DBSCAN can constantly detect

the inner circle; Cluster EBGAN can detect the inner circle in nearly 80% of the runs; while K-means, agglomerative, and Cluster GAN failed to detect the inner circle. The comparison with Cluster GAN indicates that Cluster EBGAN has made a significant improvement in GAN training.

Iris This is a classical clustering example. It contains the data for 50 flowers from each of three species - Setosa, Versicolor and Virginica. The dataset is available at `https://archive.ics.uci.edu/ml/datasets/iris`, which gives the measurements of the variables sepal length and width and petal length and width for each of the flowers. Table 2 summarizes the performance of different methods on the dataset. Other than ARI, the cluster purity is also calculated as a measure of the quality of clusters. Suppose that the data consists of K clusters and each cluster consists of n_k observations denoted by $\{x_k^j\}_{j=1}^{n_k}$. If the data were grouped into M clusters, then the cluster purity is defined by

$$\frac{\sum_{k=1}^{K} \max\{\sum_{j=1}^{n_k} 1(\mathcal{E}_2(x_k^j) = l) : l = 1, 2, \ldots, M\}}{n_1 + \cdots + n_K},$$

which measures the percentage of the samples being correctly clustered. Both the measures, ARI and cluster purity, have been used in Mukherjee et al. (2019) for assessing the performance of Cluster GAN. The comparison

shows that Cluster EBGAN significantly outperforms Cluster GAN and classical clustering methods in both ARI and cluster purity.

Seeds The dataset is available at `https://archive.ics.uci.edu/ml/datasets/seeds`. The examined group comprised kernels belonging to three different varieties of wheat: Kama, Rosa and Canadian, 70 elements each, randomly selected for the experiment. Seven geometric parameters of wheat kernels were measured, including area, perimeter, compactness, length, width, asymmetry coefficient, and length of kernel groove. Table 2 summarizes the performance of different methods on the dataset. The comparison indicates that Cluster EBGAN significantly outperforms others in both ARI and cluster purity. For this dataset, DBSCAN is not available any more, as performing density estimation in a 7-dimensional space is hard.

MNIST The MNIST dataset consists of 70,000 images of digits ranging from 0 to 9. Each sample point is a 28×28 grey scale image. Figure S8 compares the images generated by Cluster GAN and Cluster EBGAN, each representing the best result achieved by the corresponding method in 5 independent runs. It is remarkable that Cluster EBGAN can generate all digits from 0 to 9 and there is no confusion of digits between different generators. However, Cluster GAN failed to generate the digit 1 and con-

fused the digits 4 and 9. Table 2 summarizes the performance of different methods, which indicates again the superiority of the Cluster EBGAN over Cluster GAN and classical nonparametric methods.

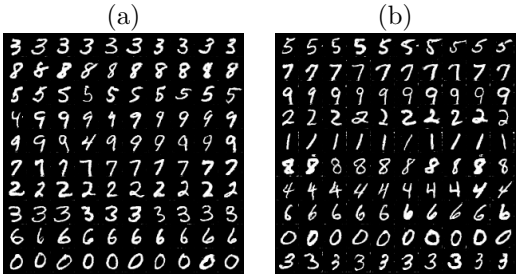


Figure S8: Images generated by (a) Cluster GAN and (b) Cluster EBGAN, where each row corresponds to a different z_c index vector.

S4 Experimental Settings

In all training with the Adam algorithm (Kingma et al., 2015), we set the tuning parameters $(\alpha_1, \alpha_2) = (0.5, 0.999)$. In this paper, all the deep convolutional GANs (DCGANs) were trained using the Adam algorithm.

For Algorithm 1 (of the main text), the step size is chosen in the form $w_t = c_1(t + c_2)^{-\zeta_1}$, and the momentum smoothing factor α is re-denoted by α_1 in Tables S3-S5. A constant learning rate ϵ and a constant momentum biasing factor $\rho = 1$.

S4.1 A Gaussian Example

Table S2 gives the parameter settings of GAN and EBGAN for the Gaussian example.

Table S2: Parameter settings for the 2D Gaussian dataset with $\phi_3(D) = log(D)$

Method	Learning rate		α_1	α_2	ρ
	Discriminator(ω_t)	Generator(ϵ_t)			
GAN	0.00002	0.0002	0.5	0.999	
EBGAN	$(c_1, c_2, \zeta_1) = (1, 1000, 0.75)$	0.01	0.9		1

S4.2 A Mixture Gaussian Example

Tables S3 and S4 give the parameter settings of different methods for the minimax and non-saturating cases, respectively. For EBGAN, we set $\tau = 0.01$.

Table S3: Parameter settings for the synthetic dataset: the minimax case with $\phi_3(D) = -\log(1 - D)$

Method	Learning rate		α_1	α_2	ρ	λ(Lipshitz)
	Discriminator	Generator				
GAN	0.0002	0.0002	0.5	0.999		
BGAN	0.001	0.001	0.9			
Probgan	0.0005	0.0005	0.5			
EBGAN	$(c_1, c_2, \zeta_1) = (1, 1000, 0.75)$	0.5	0.9		1	
Lipshitz-GAN	0.0002	0.0002	0.5	0.999		5
Lipshitz-EBGAN	$(c_1, c_2, \zeta_1) = (1, 1000, 0.75)$	0.5	0.9		1	5

Table S4: Parameter settings for the synthetic dataset: the non-saturating case with $\phi_3(D) = \log(D)$

Method	Learning rate		α_1	α_2	ρ	λ(Lipshitz)
	Discriminator	Generator				
GAN	0.0002	0.0002	0.5	0.999		
BGAN	0.001	0.001	0.9			
Probgan	0.0005	0.0005	0.9			
EBGAN	$(c_1, c_2, \zeta_1) = (1, 1000, 0.75)$	0.5	0.9		1	
Lipshitz-GAN	0.0002	0.0002	0.5	0.999		5
Lipshitz-EBGAN	$(c_1, c_2, \zeta_1) = (1, 1000, 0.75)$	0.5	0.9		1	5

S4.3 Fashion MNIST

The network structures of all models are typical DCGAN style. We set the mini-batch size to 300, set the total number of epochs to 200, and set the dimension of z_n to 10. For training the inception model, we used Adam with a learning rate of 0.0003, $(\alpha_1, \alpha_2) = (0.9, 0.999)$, a mini-batch size of 50, and 5 epochs. After training, the prediction accuracy on the test data set was 0.9304. Table S5 gives the parameter settings used by different methods. And we set $\zeta_2 = \frac{1}{40}$ for EBGAN referring to Kim et al. (2022). In addition, we set $\tau = 0.001$ for EBGAN, and set $k_g = 10$ for EBGAN, BGAN and ProbGAN.

S4.4 HMNIST

We set the latent dimension as 20, and use the normal prior $N(0, \frac{1}{80})$ on 7 generator parameters with temperature $\tau = 0.001$, with 200 batch size.

Table S5: Parameter settings for the Fashion MNIST

Method	Learning rate		α_1	α_2	ρ
	Discriminator	Generator			
GAN	0.0002	0.0002	0.5	0.999	
BGAN	0.005	0.005	0.9		
ProbGAN	0.005	0.005	0.9		
EBGAN-KL	$(c_1, c_2, \zeta_1) = (0.5, 250, 1)$ with Adam	0.01	0.9		1
EBGAN-Gaussian	$(c_1, c_2, \zeta_1) = (1, 500, 1)$ with Adam	0.01	0.9		1

Table S6: Model structure of EBGAN for Fashion MNIST

Generator	Discriminator
4× 4 conv, 512 stride 2 ReLU	4× 4 conv, 64 stride 2 pad 1 LReLU
3× 3 conv, 256 stride 2 pad 1 ReLU	4× 4 conv, 128 stride 2 pad 1 LReLU
4× 4 conv, 128 stride 2 pad 1 ReLU	3× 3 conv, 256 stride 2 pad 1 LReLU
4×4 upconv 64 stride 2 pad 1 Tanh	4× 4 conv, 512 stride 2 LReLU

Other parameter settings and the model structure are given in Table S7 and Table S8, respectively.

Table S7: Parameter settings for the HMNIST

Method	Learning rate		α_1	α_2	ρ
	Discriminator	Generator			
GAN	0.0002	0.0002	0.5	0.999	
EBGAN-Gaussian	$(c_1, c_2, \zeta_1) = (0.05, 250, 1)$ with Adam	0.001	0.9		1

S4.5 Conditional independence test

Simulated Data The network structures of all models we used are the same as in Bellot and van der Schaar (2019). In short, the generator net-

Table S8: Model structure of EBGAN for HMNIST

Generator	Discriminator
4× 4 conv, 512 stride 2 ReLU	4× 4 conv, 64 stride 2 pad 1 LReLU
3× 3 conv, 256 stride 2 pad 1 ReLU	4× 4 conv, 128 stride 2 pad 1 LReLU
4× 4 conv, 128 stride 2 pad 1 ReLU	3× 3 conv, 256 stride 2 pad 1 LReLU
4×4 upconv 64 stride 2 pad 1 Tanh	4× 4 conv, 512 stride 2 LReLU

work has a structure of $(d + d/10) - (d/10) - 1$ and the discriminator network has a structure of $(1 + d) - (d/10) - 1$, where d is the dimension of the confounding vector Z. All experiments for GCIT were implemented with the code given at `https://github.com/alexisbellot/GCIT/blob/master/GCIT.py`. For the functions ϕ_1, ϕ_2 and ϕ_3, the nonsaturating settings were adopted, i.e., we set $(\phi_1, \phi_2, \phi_3) = (\log x, \log(1 - x), \log x)$. For both cases of the synthetic data, EBGAN was run with a mini-batch size of 64, Adam optimization was used with learning rate 0.0001 for discriminator. A prior $p_g = N(0, 100I_p)$, p for dimension of parameters, and a constant learning rate of 0.005 were used for the generator. Lastly, we set $\tau = 1$. Each run consisted of 1000 iterations for case 1, case 2 and case 3. KCIT and RcoT were run by R-package at `https://github.com/ericstrobl/RCIT`.

CCLE Data For the CCLE dataset, EBGAN was run for 1000 iterations and $(c_1, c_2, \eta_1, \alpha_1) = (1, 1000, 0.75, 0.9)$ was used. Other parameters were set as above.

S4.6 Nonparametric Clustering

Two Circle We set the dimension of z_n to 3, set $(\beta_n, \beta_c) = (0.1, 0.1)$, set the mini-batch size to 500, and set a constant learning rate of 0.05 with $\tau = 1$ for the generator. For optimization of the discriminator, we used Adam and set $(\alpha_1, \alpha_2) = (0.5, 0.9)$ with a constant learning rate of 0.1. The total number of epochs was set to 2000.

Table S9: Model structure of Cluster-GAN and Cluster-EBGAN for two-circle data

Generator	Encoder	Discriminator
FC 20 LReLU	FC 20 LReLU	FC 30 LReLU
FC 20 LReLU	FC 20 LReLU	FC 30 LReLU
FC 2 linear Tanh	FC 5 linear	FC 1 linear

Iris For the iris data, we used a simple feed-forward network structure for Cluster GAN and Cluster EBGAN. We set the dimension of z_n to 20, set $(\beta_n, \beta_c) = (10, 10)$, set the mini-batch size to 32, and set a constant learning rate of 0.01 for the generator with $\tau = 1$. For optimization of the discriminator, we used Adam and set $(\alpha_1, \alpha_2) = (0.5, 0.9)$ with a learning rate of 0.0001. The hyperparameters of Cluster-GAN is set to the default values.

Seeds For the seeds data, we used a simple feed-forward network structure for Cluster-GAN and Cluster-EBGAN. We set the dimension of z_n to 20,

Table S10: Model structure of Cluster-GAN and Cluster-EBGAN for Iris

Generator	Encoder	Discriminator
FC 5 LReLU	FC 5 LReLU	FC 5 LReLU
FC 5 LReLU	FC 5 LReLU	FC 5 LReLU
FC 4 linear Sigmoid	FC 23 linear	FC 1 linear

set $(\beta_n, \beta_c) = (5, 5)$, set the mini-batch size to 128, and set a constant learning rate of 0.01 for generator with $\tau = 0.0001$. For optimization of the discriminator, we used Adam and set $(\alpha_1, \alpha_2) = (0.5, 0.9)$ with a learning rate of 0.005. The hyperparameters of Cluster-GAN is set to the default values.

Table S11: Model structure of Cluster-GAN and Cluster-EBGAN for Seeds

Generator	Encoder	Discriminator
FC 20 LReLU	FC 20 LReLU	FC 100 LReLU
FC 20 LReLU	FC 20 LReLU	FC 100 LReLU
FC 7 linear Tanh	FC 23 linear	FC 1 linear

MNIST For Cluster GAN, our implementation is based on the code given at `https://github.com/zhampel/clusterGAN`, with a a small modification on Encoder. The Structures of the generator, encoder and discriminator are given as follow. Cluster GAN was run with the same parameter setting as given in the original work Mukherjee et al. (2019).

For Cluster EBGAN, to accelerate computation, we used the parameter sharing strategy as in Hoang et al. (2018), where all generators share the

Table S12: Model structure of ClusterGAN for MNIST data

Generator	Encoder	Discriminator
FC 1024 ReLU BN	4× 4 conv, 64 stride 2 LReLU	4× 4 conv, 64 stride 2 LReLU
FC $7 \times 7 \times 128$ ReLU BN	4× 4 conv, 128 stride 2 LReLU	4× 4 conv, 64 stride 2 LReLU
4×4 upconv 64 stride 2 ReLU BN	4× 4 conv, 256 stride 2 LReLU	FC1024 LReLU
4×4 upconv 1 stride 2 Sigmoid	FC 1024 LReLU	FC 1 linear
	FC 40	

parameters except for the first layer. We set the dimension of z_n to 5, $(c_1, c_2, \eta_1, \alpha_1) = (40, 10000, 0.75, 0.9)$, set the mini-batch size to 100, and set a constant learning rate of 0.005 for the generator. For the functions ϕ_1, ϕ_2 and ϕ_3, the non-saturating settings were adopted, i.e., we set $(\phi_1, \phi_2, \phi_3) = (\log x, \log(1-x), \log x)$.

Table S13: Model structure of ClusterEBGAN for MNIST simulation

Generator	Encoder	Discriminator
4× 4 conv, 512 stride 2 ReLU	4× 4 conv, 64 stride 2 LReLU	4× 4 conv, 64 stride 2 LReLU
3× 3 conv, 128 stride 2 pad 1 ReLU	4× 4 conv, 128 stride 2 LReLU	4× 4 conv, 64 stride 2 LReLU
4× 4 conv, 64 stride 2 pad 1 ReLU	4× 4 conv, 256 stride 2 LReLU	FC1024 LReLU
4×4 upconv 1 stride 2 pad 1 Sigmoid	FC 1024 LReLU	FC 1 linear
	FC 30	

Bibliography

Andrieu, C., E. Moulines, and P. Priouret (2005). Stability of stochastic approximation under verifiable conditions. *SIAM Journal on Control and Optimization 44*(1), 283–312.

Bellot, A. and M. van der Schaar (2019). Conditional independence testing using generative adversarial networks. In *NeurIPS*, pp. 2202–2211.

Benveniste, A., M. Métivier, and P. Priouret (1990). *Adaptive Algorithms and Stochastic Approximations*. Springer.

Berrett, T., Y. Wang, R. Barber, and R. Samworth (2019, 10). The conditional permutation test for independence while controlling for confounders. *Journal of the Royal Statistical Society: Series B (Statistical Methodology) 82*.

Candès, E., Y. Fan, L. Janson, and J. Lv (2018). Panning for gold: 'model-x' knockoffs for high dimensional controlled variable selection. *Journal of the Royal Statistical Society: Series B (Statistical Methodology) 80*(3), 551–577.

Chen, C., N. Ding, and L. Carin (2015). On the convergence of stochastic gradient mcmc algorithms with high-order integrators. In *Advances in Neural Information Processing Systems*, pp. 2278–2286.

Chen, H. and Y. Zhu (1986). Stochastic approximation procedures with randomly varying truncations. *Science in China Series A-Mathematics, Physics, Astronomy & Technological Science 29*(9), 914–926.

Deng, W., X. Zhang, F. Liang, and G. Lin (2019). An adaptive empirical bayesian method for sparse deep learning. *NeurIPS 2019*.

Doran, G., K. Muandet, K. Zhang, and B. Schölkopf (2014). A permutation-based kernel conditional independence test. In *Proceedings of the Thirtieth Conference on Uncertainty in Artificial Intelligence*, UAI'14, Arlington, Virginia, USA, pp. 132–141. AUAI Press.

Dragomir, S., M. Scholz, and J. Sunde (2000). Some upper bounds for relative entropy and applications. *Computers & Mathematics with Applications 39*(9), 91–100.

Gao, X., M. Gürbüzbalaban, and L. Zhu (2021). Global convergence of stochastic gradient hamiltonian monte carlo for nonconvex stochastic optimization: Nonasymptotic performance bounds and momentum-based acceleration. *Operations Research*.

Garnett, M. J., E. J. Edelman, S. J. Heidorn, C. D. Greenman, A. Dastur, K. W. Lau, P. Greninger, I. R. Thompson, X. Luo, J. Soares, Q. Liu, F. Iorio, D. Surdez, L. Chen, R. J. Milano, G. R. Bignell, A. T. Tam, H. Davies, J. A. Stevenson, S. Barthorpe, S. R. Lutz, F. Kogera, K. Lawrence, A. McLaren-Douglas, X. Mitropoulos, T. Mironenko, H. Thi, L. Richardson, W. Zhou, F. Jewitt, T. Zhang, P. O'Brien, J. L. Boisvert, S. Price,

W. Hur, W. Yang, X. Deng, A. Butler, H. G. Choi, J. W. Chang, J. Baselga, I. Stamenkovic, J. A. Engelman, S. V. Sharma, O. Delattre, J. Saez-Rodriguez, N. S. Gray, J. Settleman, P. A. Futreal, D. A. Haber, M. R. Stratton, S. Ramaswamy, U. McDermott, and C. H. Benes (2012, March). Systematic identification of genomic markers of drug sensitivity in cancer cells. *Nature 483*(7391), 570—575.

Goodfellow, I., J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio (2014). Generative adversarial nets. *NIPS*, 2672–2680.

Hoang, Q., T. D. Nguyen, T. Le, and D. Phung (2018). MGAN: Training generative adversarial nets with multiple generators. In *ICLR*.

Kass, R. E., L. Tierney, and J. B. Kadane (1990). The validity of posterior expansions based on Laplace's method. In S. Geisser, J. S. Hodges, S. J. Press, and A. ZeUner (Eds.), *Bayesian and likelihood methods in statistics and econometrics: essays in honor of George A. Barnard*, Volume 7, pp. 473–488. Amsterdam: North Holland.

Kim, S., Q. Song, and F. Liang (2022). Stochastic gradient langevin dynamics with adaptive drifts. *Journal of Statistical Computation and Simulation 92*(2), 318–336.

Kingma, D. P., and J. L. Ba (2015). Adam: a method for stochastic optimization. In *International Conference on Learning Representations*.

Lauritzen, S. (1996). *Graphical Models*. Clarendon Press.

Li, C. and X. Fan (2019, 12). On nonparametric conditional independence tests for continuous variables. *Wiley Interdisciplinary Reviews: Computational Statistics 12*.

Liang, F., Q. Song, and P. Qiu (2015). An equivalent measure of partial correlation coefficients for high dimensional gaussian graphical models. *Journal of the American Statistical Association 110*, 1248–1265.

Lopez-Paz, D., P. Hennig, and B. Schölkopf (2013). The randomized dependence coefficient. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger (Eds.), *Advances in Neural Information Processing Systems*, Volume 26, pp. 1–9. Curran Associates, Inc.

Mukherjee, S., H. Asnani, E. Lin, and S. Kannan (2019). Clustergan: Latent space clustering in generative adversarial networks. In *AAAI*, pp. 4610–4617. AAAI Press.

Pearl, J. (2009, 01). Causal inference in statistics: An overview. *Statistics Surveys 3*, 96–146.

Raginsky, M., A. Rakhlin, and M. Telgarsky (2017). Non-convex learning via stochastic gradient langevin dynamics: a nonasymptotic analysis. In *Conference on Learning Theory*, pp. 1674–1703. PMLR.

Rand, W. M. (1971). Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical Association 66*(336), 846–850.

Salimans, T., I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen (2016). Improved techniques for training gans. In *NIPS*, pp. 2234–2232.

Sen, R., A. T. Suresh, K. Shanmugam, A. G. Dimakis, and S. Shakkettai (2017). Model-powered conditional independence test. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, Red Hook, NY, USA, pp. 2955–2965. Curran Associates Inc.

Singleton, K. R., L. Crawford, E. Tsui, H. E. Manchester, O. Maertens, X. Liu, M. V. Liberti, A. N. Magpusao, E. M. Stein, J. P. Tingley, D. T. Frederick, G. M. Boland, K. T. Flaherty, S. J. McCall, C. Krepler, K. Sproesser, M. Herlyn, D. J. Adams, J. W. Locasale, K. Cichowski, S. Mukherjee, and K. C. Wood (2017). Melanoma therapeutic strate-

gies that select against resistance by exploiting myc-driven evolutionary convergence. *Cell Reports 21*(10), 2796 – 2812.

Spirtes, P., C. Glymour, and R. Scheines (1993). *Causation, prediction and search*. New York: Springer.

Strobl, E. V., K. Zhang, and S. Visweswaran (2019). Approximate kernel-based conditional independence tests for fast non-parametric causal discovery. *Journal of Causal Inference 7*(1), 20180017.

Tripathi, R., Z. Liu, A. Jain, A. Lyon, C. Meeks, D. Richards, J. Liu, D. He, C. Wang, M. Nespi, A. Rymar, P. Wang, M. Wilson, and R. Plattner (2020, 11). Combating acquired resistance to mapk inhibitors in melanoma by targeting abl1/2-mediated reactivation of mek/erk/myc signaling. *Nature Communications 5463*.

Wang, Z., A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli (2004). Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing 13*, 600–612.

Whye, T., H. ThieryAlexandre, and J. VollmerSebastian (2016). Consistency and fluctuations for stochastic gradient langevin dynamics. *Journal of Machine Learning Research*.

Xu, Q., G. Huang, Y. Yuan, C. Guo, Y. Sun, F. Wu, and K. Q. Weinberger

(2018). An empirical study on evaluation metrics of generative adversarial networks. *ArXiv abs/1806.07755*.

Zhang, K., J. Peters, D. Janzing, and B. Schölkopf (2012). Kernel-based conditional independence test and application in causal discovery. In *Proceedings of the Twenty-Seventh Conference on Uncertainty in Artificial Intelligence*, UAI'11, Arlington, Virginia, USA, pp. 804–813. AUAI Press.

Zhang, Q., S. Filippi, S. Flaxman, and D. Sejdinovic (2017). Feature-to-feature regression for a two-step conditional independence test. In *UAI*.