Generative adversarial learning with optimal input dimension and its adaptive generator architecture

---

Huazhen Lin

New Cornerstone Science Laboratory,

Center of Statistical Research and School of Statistics,

Southwestern University of Finance and Economics

This is joint work with Zhiyao Tan and Ling Zhou

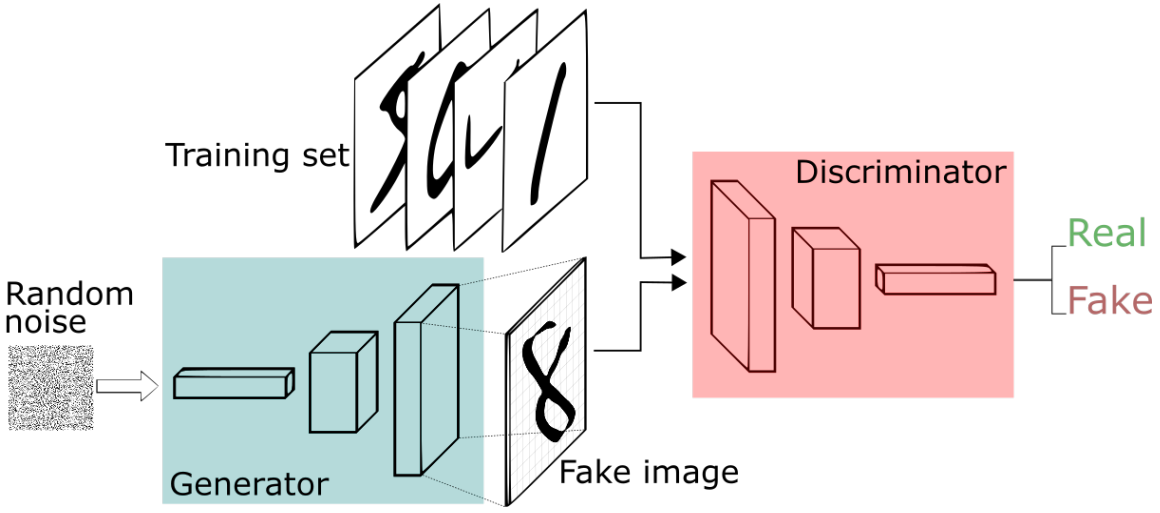
2024 JCSD in China

Outline

- **Background**
- **Related works on generalization error of GANs**
- **Theoretical Results**
- **Method**
- **Selection consistency**
- **Implementation**
- **Experiments**
- **Concluding Remarks**

Background

Generative adversarial networks (GANs) is a class of deep generative models.



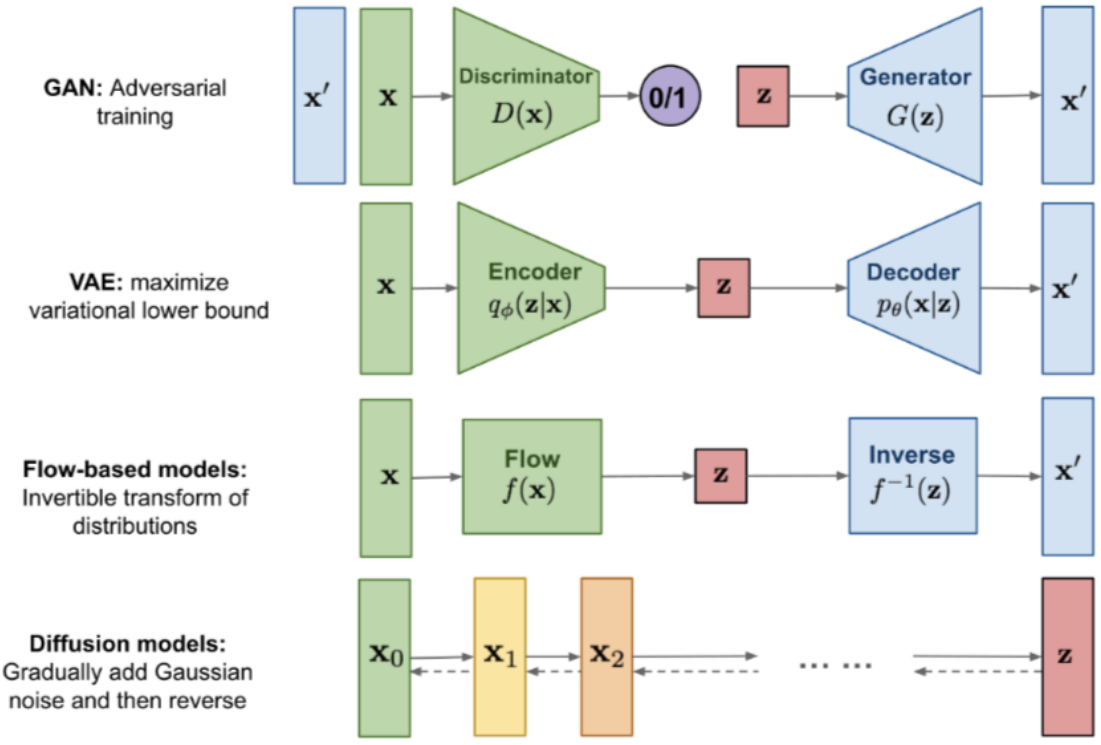
- **Purpose: Learn the data distribution and generate new samples.**

- **The discriminator f and generator g train adversarially through**

$$
\begin{aligned}
(f^*, g^*) &= \arg\min_{g \in \mathcal{G}} \max_{f \in \mathcal{F}} \mathcal{L}(f, g), \\
\mathcal{L}(f, g) &= \mathbb{E}_{x \sim \mu}[f(x)] - \mathbb{E}_{z \sim \nu}[f(g(z))],
\end{aligned} \tag{1}
$$

- **ν is an easy-to-sample source distribution, such as uniform or Gaussian distribution, of dimension d;**
- **μ is the target distribution for high-dimensional data of dimension D;**
- **$\mathcal{G}$ and $\mathcal{F}$, the generator and the discriminator classes, are parameterized by neural networks.**

Several common generative models



The comparison of generative models

- **VAEs assume a normal distribution, leading to blurriness due to the induced L_2 loss in reconstrution. In contrast, GANs use adversarial loss which is distribution-guided, producing higher-quality images.**
- **Flow-based models necessitate complex, invertible transformations.**
- **Diffusion models generate images through a slow, multi-step process, which can be computationally expensive.**

The underlying principle of GANs

- **Real-world datasets possess low-dimensional intrinsic structures (Arjovsky and Bottou, 2017; Dahal et al., 2022).**
- **GANs hence create high-dimensional data of dimension D from low-dimensional input variables z with $d \ll D$.**

The problems of GANs

- However, both the theoretical and practical understanding of d remain unclear.
- In particular, given observed data and an easy-to-sample distribution, how the generalization error of a GAN depends on the input dimension d.

Outline

- Background
- **Related works on generalization error of GANs**
- Theoretical Results
- Method
- Selection consistency
- Implementation
- Experiments
- Concluding Remarks

Generalization error of GANs

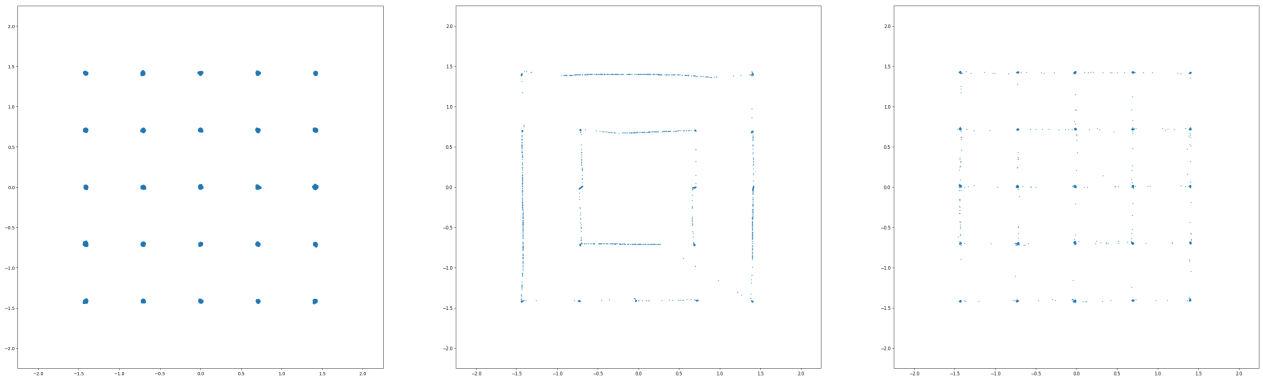
The generalization error of GANs can be decomposed into three main components:

- **generator approximation error**
- **discriminator approximation error**
- **statistical error.**

Generator approximation error

- **When $d = D$ (Bailey and Telgarsky, 2018; Chen et al., 2020), the established approximation rate of $\mathcal{O}(W)^{-\mathcal{O}(L/D)}$ or $\mathcal{O}(W^{-\frac{1}{D}})$ produces the curse of dimensionality.**
- **When $d = 1$ (Perekrestenko et al., 2020; Yang et al., 2022; Huang et al., 2022), the approximation error for the empirical target distribution vanishes. This approximation ignores the regularity or smoothness of the target distribution, resulting in poor generalization.**
- **These above results are derived from a fixed d.**

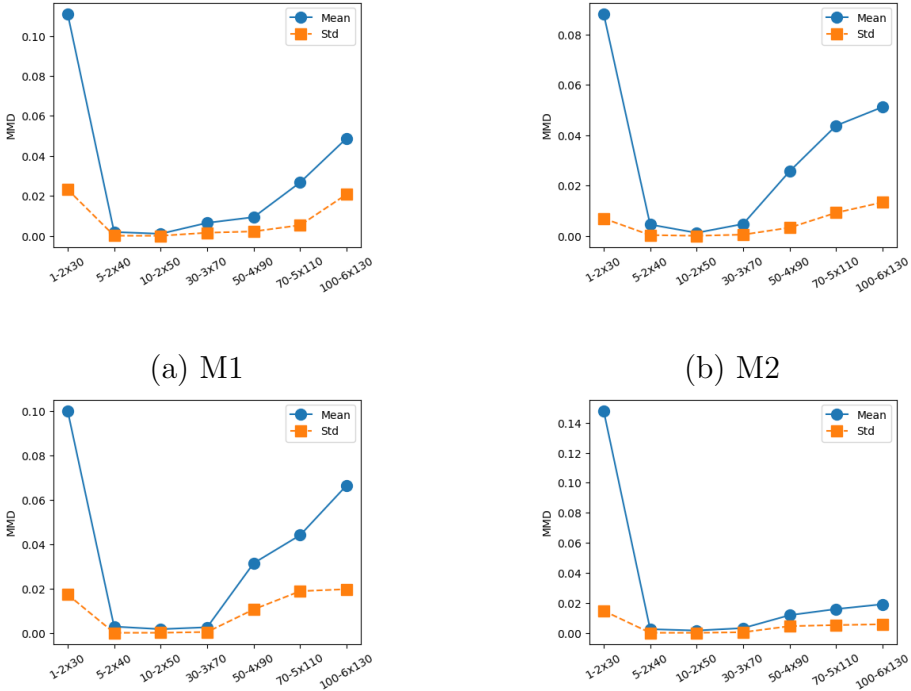
Numerical result-I



(c) (d) (e)

FIGURE 1. A practical verification of Theorem 5 in Huang et al. (2022) on the GRID dataset. (a) shows 1000 samples from the GRID dataset, (b) shows 1000 samples generated by WGAN-GP with an input dimension d=1 and (c) shows 1000 samples generated by WGAN-GP with an input dimension d=2.

Numerical result-II



(a) M1 (b) M2

(c) M3 (d) M4

FIGURE 2. The mean and standard deviation of Maximum Mean Discrepancy (MMD) of SNGANs (Miyato et al., 2018).

Discriminator approximation error

- **Rely on the observed data and is independent of d.**
- **$\mathcal{O}((WL)^{-2\beta/D})$ (Jiao et al., 2023). This rate can be improved to $\mathcal{O}((WL)^{-2\beta/d^*})$, where $d^* \leq D$ is the intrinsic dimension of data.**

Statistical error

- $O_p(n^{-\frac{\beta}{2\beta+D}})$ **when** $d = D$ (Chen et al., 2020).
- $O_p(\max(n^{-\frac{\beta}{d^*}}, n^{-\frac{1}{2}}))$ (Schreuder et al., 2021; Huang et al., 2022).
- **These above results are derived from either a fixed** d **or are independent of** d**.**

To the best of our knowledge, these is a lack of investigation on how does the generalization error of GANs vary with the input dimension d**.**

The aims of this work

- **How does the generalization error of GANs vary with the input dimension d?**
- **In theory, whether there is an optimal input dimension d that minimizes the generalization error?**
- **If it exists, how identify?**
- **What network architecture should the generator g have for a given d?**

Our works

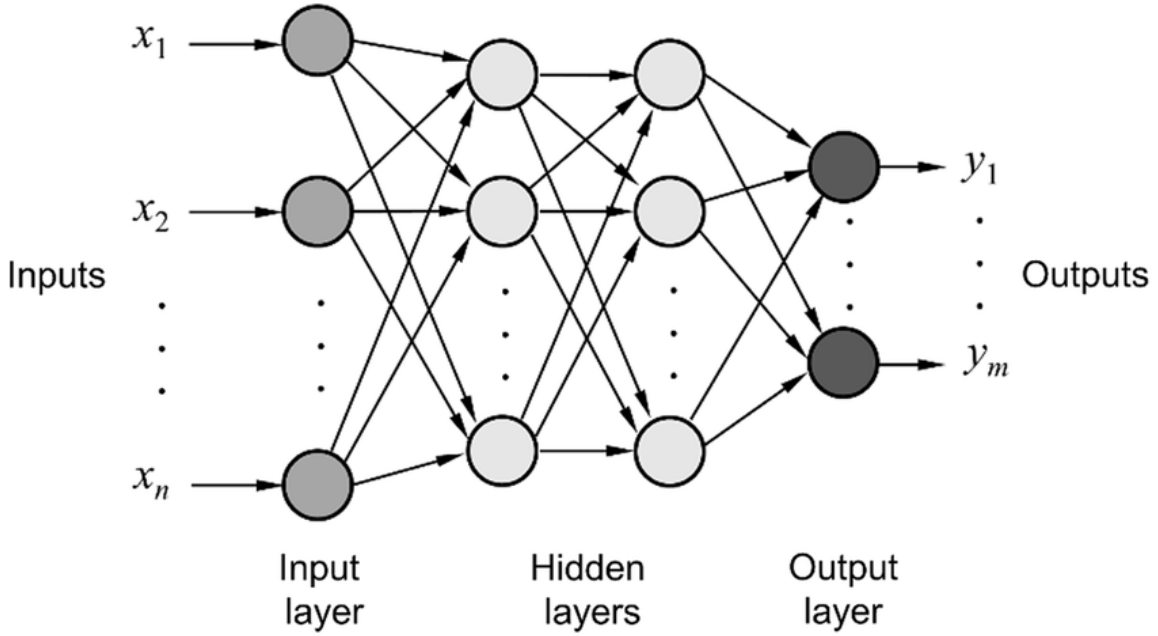
- **We first provide both theoretical and practical evidence to validate the existence of the optimal input dimension (OID) that minimizes the generalization error.**
- **We introduce a novel framework called generalized GANs (G-GANs) to identify the OID, along with its corresponding generator network.**
- **We provide the consistent selection theory of the identified input dimension and generator network.**

Outline

- Background
- Related works on generalization error of GANs
- **Theoretical Results**
- Method
- Selection consistency
- Implementation
- Experiments
- Concluding Remarks

Feedforward neural network (FNN)

We parameterize generator class $\mathcal{G}$ and discriminator class $\mathcal{F}$ by FNN with ReLU activation function.



A ReLU neural network with L hidden layers is a collection of mappings $\phi : \mathbb{R}^{N_0} \to \mathbb{R}^{N_{L+1}}$ of the form:

$$\phi(x) = T_L \circ \sigma \circ T_{L-1} \circ \cdots \circ \sigma \circ T_0(x).$$

- **$\phi_1 \circ \phi_2(x) := \phi_1(\phi_2(x))$ represents the composition of two functions. $\sigma(x) := \max(x, 0)$ is the ReLU function. $T_l(x) := A_l x + c_l$ is an affine transformation.**
- **$W = \max\{N_1, N_2, ..., N_L\}$, L and $\mathcal{S} = \sum_{i=0}^{L} N_{i+1} \times (N_i + 1)$ are width, depth and size (parameter counts) of ϕ, where N_l is the number of neurons in layer l.**

Denote $\mathcal{NN}(W, L, \mathcal{S}, \mathcal{B})$ the set of ϕ with width W, depth L, size $\mathcal{S}$ and $\|\phi\|_\infty \leq \mathcal{B}$ for some $0 < \mathcal{B} < \infty$, where $\|\phi\|_\infty$ is the supnorm of the function ϕ.

GANs with finite samples

- **Based on a finite collection of samples $X_1, ..., X_n \overset{i.i.d}{\sim} \mu$ and $Z_1, ..., Z_m \overset{i.i.d}{\sim} \nu$, we estimate g by**

$$\hat{g} = \arg\min_{g \in \mathcal{G}} \max_{f \in \mathcal{F}} \hat{\mathcal{L}}(f, g) := \arg\min_{g \in \mathcal{G}} \max_{f \in \mathcal{F}} \left\{ \frac{1}{n} \sum_{i=1}^{n} f(X_i) - \frac{1}{m} \sum_{j=1}^{m} (f(g(Z_j))) \right\}. \tag{2}$$

- **The error of $\hat{g}$ is evaluated by the integral probability metric (IPM) (Müller, 1997) with respect to the evaluation class $\mathcal{H}$:**

$$d_{\mathcal{H}}(\hat{g}_{\#}\nu, \mu) = \max_{h \in \mathcal{H}} \mathbb{E}_{x \sim \mu}[h(x)] - \mathbb{E}_{z \sim \nu}[h(\hat{g}(z))], \tag{3}$$

where $\hat{g}_{\#}\nu(\mathcal{A}) := \nu(\hat{g}^{-1}(\mathcal{A}))$ for a measurable set $\mathcal{A}$.

Theoretical results

Definition 2.1. **A mapping class** $\mathcal{G}_0 := \underset{d \le D}{\cup} \{g_0 : [0,1]^d \to [0,1]^D, g_0 \in \mathcal{H}^{\beta_1}([0,1]^d)\}$ **such that** $\forall g_0 \in \mathcal{G}_0$, $\mu = g_{0\#}(\nu)$.

Definition 2.2[Minimal input dimension]. **We write** g^d **as the** d**-dimensional function and** ν^d **as the source distribution of the** d**-dimension. The minimal input dimension (MID) is defined as**

$$d_0 := \min_d \{d \mid \mu = g_{0\#}^d(\nu^d), g_0 \in \mathcal{G}_0\},$$

where the minimum is taken among all d**s such that** (g^d, ν^d) **can exactly generate the target distribution.**

Theorem 2.1. **Suppose the target distribution** $\mu = g_{0\#}(\nu)$ **with** $g_0 \in \mathcal{H}^{\beta_1}([0,1]^d)$, **the evaluation class is** $\mathcal{H}^{\beta_2}([0,1]^D)(\beta_2 \geq 1)$. **Then, there exists a generator class** $\mathcal{G} = \{g : \mathbb{R}^d \to \mathbb{R}^D \mid g \in \mathcal{NN}(W_\mathcal{G}, L_\mathcal{G}, \mathcal{S}_\mathcal{G}, \mathcal{B}_\mathcal{G})\}$ **with**

$$W_\mathcal{G} L_\mathcal{G} \preceq n^{\frac{d}{2(2\beta_1+d)}},$$

and a discriminator class $\mathcal{F} = \{f : \mathbb{R}^D \to \mathbb{R} \mid f \in \mathcal{NN}(W_\mathcal{F}, L_\mathcal{F}, \mathcal{S}_\mathcal{F}, \mathcal{B}_\mathcal{F})\}$ **with**

$$W_\mathcal{F} L_\mathcal{F} \preceq n^{\frac{D}{2(2\beta_2+D)}},$$

so that GAN estimator (2) **satisfies**

$$\begin{aligned} d_{\mathcal{H}^{\beta_2}}(\hat{g}_\# \nu, \mu) = O_p \Bigg(& n^{\frac{-\beta_2}{2\beta_2+D}} + \{ n^{\frac{-\beta_1}{2\beta_1+d}} + \inf_{\bar{g}^d \in \mathcal{H}^{\beta_1}([0,1]^d)} d_\mathcal{F}(\bar{g}^d_\# \nu^d, g^{d_0}_{0\#} \nu^{d_0}) \} \\ & + \{ n^{\frac{-\beta_1}{2\beta_1+d}} \log^2 n + n^{\frac{-\beta_2}{2\beta_2+D}} \log^2 n \\ & + m^{\frac{-1}{2}} n^{\frac{d}{2(2\beta_1+d)}} \log^{\frac{3}{2}} n \log^{\frac{1}{2}} m + m^{\frac{-1}{2}} n^{\frac{D}{2(2\beta_2+D)}} \log^{\frac{3}{2}} n \log^{\frac{1}{2}} m \} \Bigg). \end{aligned} \tag{4}$$

- **Discriminator approximation error:** $\mathcal{O}_p(n^{\frac{-\beta_2}{2\beta_2+D}})$.
- **Generator approximation error:**

$$\mathcal{O}_p(n^{\frac{-\beta_1}{2\beta_1+d}} + \inf_{\bar{g}^d \in \mathcal{H}^{\beta_1}([0,1]^d)} d_{\mathcal{F}}(\bar{g}^d_{\#}\nu^d, g^{d_0}_{0\#}\nu^{d_0})).$$

- **Statistical error:**

$$\mathcal{O}_p(n^{\frac{-\beta_1}{2\beta_1+d}}\log^2 n + n^{\frac{-\beta_2}{2\beta_2+D}}\log^2 n + m^{\frac{-1}{2}} n^{\frac{d}{2(2\beta_1+d)}}\log^{\frac{3}{2}} n \log^{\frac{1}{2}} m + m^{\frac{-1}{2}} n^{\frac{D}{2(2\beta_2+D)}}\log^{\frac{3}{2}} n \log^{\frac{1}{2}} m).$$

- **If there exists a constant C and an intrinsic dimension d^* $(d^* \leq D)$ such that $N(\epsilon, \Omega, \|\cdot\|_\infty) \leq C\epsilon^{-d^*}$, the two terms $\mathcal{O}_p(n^{\frac{-\beta_2}{2\beta_2+D}})$ and $\mathcal{O}_p(n^{\frac{-\beta_2}{2\beta_2+D}}\log^2 n)$ can be improved to $\mathcal{O}_p(n^{\frac{-\beta_2}{2\beta_2+d^*}})$ and $\mathcal{O}_p(n^{\frac{-\beta_2}{2\beta_2+d^*}}\log^2 n)$, where $N(\epsilon, \Omega, \rho)$ is the covering number of Ω under the metric ρ with radius ϵ.**

Comparison to existing results

- Chen et al. (2020): $O_p(n^{\frac{-\beta_2}{2\beta_2+D}} \log^2 n)$.
  - **A special case of our results where $d = D$, $\beta_1 = \beta_2 = \beta$ and $m \geq n$.**
- Huang et al. (2022): $O_p(n^{\frac{-\beta}{D}} \vee n^{\frac{-1}{2}} \log n)$.
  - **The generator size, $W_{\mathcal{G}}^2 L_{\mathcal{G}} \preceq n$, is much larger than that in our proposal, i.e., $W_{\mathcal{G}} L_{\mathcal{G}} \preceq n^{\frac{d}{2(2\beta_1+d)}}$.**
  - **This approach, based on memorizing empirical data, overlooks the smoothness of the target distribution.**
  - **Their theoretical requirement $m > n^{2+2\beta/d} \log^6 n$ is impractical.**

Generalization error variation with d

- **Generalization error first decreases then increases as d increases.**
  - **Decrease when $d < d_0$:** $\inf_{\bar{g}^d \in \mathcal{H}^{\beta_1}([0,1]^d)} d_{\mathcal{F}}(\bar{g}^d_{\#}\nu^d, g^{d_0}_{0\#}\nu^{d_0})$ **remains significantly distant from zero and dominates this error and it decreases as d increases.**
  - **Increase when $d \geq d_0$:** $\inf_{\bar{g}^d \in \mathcal{H}^{\beta_1}([0,1]^d)} d_{\mathcal{F}}(\bar{g}^d_{\#}\nu^d, g^{d_0}_{0\#}\nu^{d_0})$ **vanishes and $n^{\frac{-\beta_1}{2\beta_1+d}}$ and statistical error $O_p(n^{\frac{-\beta_1}{2\beta_1+d}} \log^2 n)$ increases as d increases.**

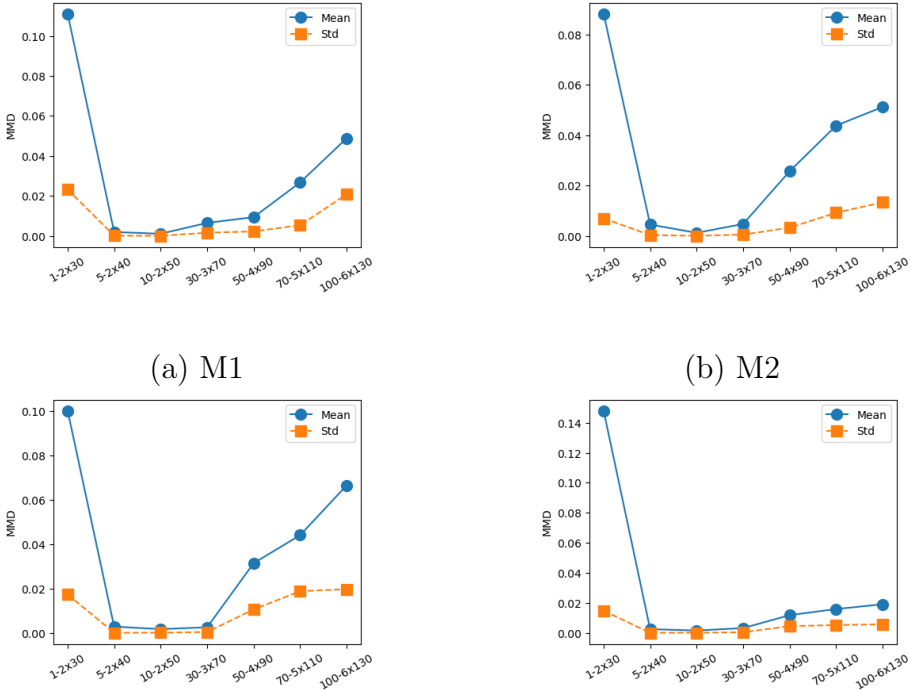
Summary: Generalization error first decreases then increases as d increases, indicating the existence of an optimal input dimension (OID).

Optimal input dimension (OID)

Corollary 2.1. Under the conditions of Theorem 2.1, if $m > n$, the GAN estimator (2) achieves the optimal error rate with $d = d_0$ on the order of

$$d_{\mathcal{H}^{\beta_2}}(\hat{g}_{\#}\nu, \mu) = O_p\left(n^{\frac{-\beta_2}{2\beta_2+D}} \log^2 n + n^{\frac{-\beta_1}{2\beta_1+d_0}} \log^2 n\right).$$

Numerical result



(a) M1 (b) M2

(c) M3 (d) M4

FIGURE 3. The mean and standard deviation of Maximum Mean Discrepancy (MMD) of SNGANs (Miyato et al., 2018).

Outline

- Background
- Related works on generalization error of GANs
- Theoretical Results
- **Method**
- Selection consistency
- Implementation
- Experiments
- Concluding Remarks

Generalized GANs (G-GANs)

A key strategy for identifying the OID is to introduce an index matrix B, leading to the generalized GANs (G-GANs) framework:

$$\mathcal{L}(f, g, \mathbf{B}) = \mathbb{E}_{x\sim\mu}[f(x)] - \mathbb{E}_{z\sim\nu}[f(g(\mathbf{B}z))].$$

- **When B is the identity matrix, the criterion $\mathcal{L}(f, g, \mathbf{B})$ is reduced to $\mathcal{L}(f, g)$, as defined in (1). Hence, the existing GANs can be regarded as a special case of the generalized GANs (G-GANs).**
- **Input dimension selection:**
  - **Impose a group sparsity penalty on the row of B.**

Objective function

$$\left(\hat{\mathbf{B}}, \hat{\theta}\right) = \underset{\mathbf{B}\in\mathcal{W}_{\mathbf{B}},\theta:g_\theta\in\mathcal{G}}{\arg\min} \max_{w:f_w\in\mathcal{F}} \left\{ \frac{1}{n}\sum_{i=1}^{n} f_w(\mathbf{X}_i) - \frac{1}{m}\sum_{j=1}^{m} f_w(g_\theta(\mathbf{B}\mathbf{Z}_j)) + \mathcal{L}_{reg}(\mathbf{B},\theta) \right\}, \tag{5}$$

where $\mathcal{L}_{reg}(\mathbf{B},\theta) = \lambda_1 M(\mathbf{B}) + \lambda_2 P(\theta) + \lambda_3 Q(\theta)$**.**

- **Adaptive group sparsity penalty on the row of B:**

  $$M(\mathbf{B}) = \sum_i \frac{1}{\|\tilde{\mathbf{B}}^{[i,:]}\|_2} \|\mathbf{B}^{[i,:]}\|_2 = \sum_i \frac{1}{\sqrt{\sum_j \tilde{\mathbf{B}}^{[i,j]2}}} \sqrt{\sum_j \mathbf{B}^{[i,j]^2}},$$

  where $\tilde{\mathbf{B}}$ is an initial estimator of B obtained by group lasso.

- **This reduction in dimensionality also shrinks the required size of the generator network architecture, which is automatically identified by architecture penalties $P(\theta)$ and $Q(\theta)$.**

Architecture penalties

A key observation: if layer l is redundant, the affine transformation $T_l(x) = A_l x + c_l$ is not necessary; that is, the information from $T_l(x)$ equals that from x.

- **Depth penalty:** $P(\theta) = \sum_{l=1}^{L_{\mathcal{G}}-1} \|A_l - \mathbf{I}\|_1 + \|c_l\|_1$.
- **Sparsity penalty:** $Q(\theta) = \|\theta\|_1$.

Outline

- Background
- Related works on generalization error of GANs
- Theoretical Results
- Method
- **Selection consistency**
- Implementation
- Experiments
- Concluding Remarks

Definition of selection consistency

- **Input dimension:**
  - $\mathbb{P}(\sum_i \mathbb{1}_{\hat{\mathbf{B}}^{[i,:]} \neq \mathbf{0}} = d_0) \to 1$ **as** $n, m \to \infty$**.**
  - $\mathbf{B} = [\mathbf{U}^\top, \mathbf{V}^\top]^\top$ **where** $\mathbf{U} \in \mathbb{R}^{d_0 \times d}$ **and** $\mathbf{V} \in \mathbb{R}^{(d-d_0)\times d}$**.**
- **Depth:**
  - $\mathbb{P}(\mathbf{dep}(\hat{\theta}) = l_\theta) \to 1$ **as** $n, m \to \infty$ **where** $\mathbf{dep}(\hat{\theta})$ **is the depth of** $\hat{\theta}$ **and** l_θ **is the minimal depth in the optimal parameter set** Θ^***.**
- **Network size:**
  - $\mathbb{P}(\|\hat{\theta}\|_0 = n_\theta) \to 1$**, as** $n, m \to \infty$ **where** n_θ **is the minimal size in the optimal parameter set** Θ^***.**

where

- $\Theta^* = \{\theta^* : (\mathbf{B}^*, \theta^*) \in \underset{\mathbf{B}\in\mathcal{W}_{\mathbf{B}},\theta:g_\theta\in\mathcal{G}}{\arg\min} \{d_{\mathcal{F}}(g_\theta \circ \mathbf{B}_{\#}\nu, \mu), \mathbf{B} = [\mathbf{U}^\top, \mathbf{0}^\top]^\top, \mathbf{U} \in \mathbb{R}^{d_0\times d}, A_l \in \mathbb{R}^{W\times W}, 1 \le l \le L_{\mathcal{G}}\}\}$ **for the parameters set of generator with the width** W **under the OID.**
- $l_\theta = \min_{l^*}\{l^* : l^* = \mathbf{dep}(\theta^*), \theta^* \in \Theta^*\}$**, where** $\mathbf{dep}(\theta^*)$ **is the depth of** θ^***.**
- $n_\theta = \min_{\breve{\theta}^*\in\breve{\Theta}^*} \|\breve{\theta}^*\|_0$ **as the minimal size of the generator under the OID and minimal depth.**
- $\breve{\Theta}^* = \{\breve{\theta}^* : \mathbf{dep}(\breve{\theta}^*) = l_\theta, \breve{\theta}^* \in \Theta^*\}$ **is the parameter set of the generator under the OID and minimal depth.**

Conditions

(C1) **For any** $\breve{\theta}_1^*, \breve{\theta}_2^* \in \breve{\Theta}^*$, $\|\breve{\theta}_1^*\|_1 \leq \|\breve{\theta}_2^*\|_1$ **implies** $\|\breve{\theta}_1^*\|_0 \leq \|\breve{\theta}_2^*\|_0$.

(C2) **For any** $\tilde{\theta}_1^*, \tilde{\theta}_2^* \in \tilde{\Theta}^*$, **there exists a constant** $M_b < \infty$ **such that** $\|\tilde{\theta}_1^* - \tilde{\theta}_2^*\|_2 \leq M_b$ **where** $\tilde{\Theta}^* = \{\tilde{\theta}^* : (\tilde{\mathbf{B}}^*, \tilde{\theta}^*) \in \underset{\mathbf{B} \in \mathcal{W}_{\mathbf{B}}, \theta : g_\theta \in \mathcal{G}}{\arg\min} \; d_{\mathcal{F}}(g_\theta \circ \mathbf{B}_{\#}\nu, \mu)\}$.

- **Condition (C1) requires L_1 penalty can approximate L_0 penalty, whihc is easy to hold due to the relatively small magnitudes of all the neural network parameters.**
- **Condition (C2) is a relaxation of the assumption that the L_1 norm of any parameter is bounded, a condition frequently required in the neural network literature.**

Selection and estimation consistency

Theorem 3.1. **Considering that the generator class** $\mathcal{G} = \{g : \mathbb{R}^d \to \mathbb{R}^D \mid g \in \mathcal{NN}(W_{\mathcal{G}}, L_{\mathcal{G}}, \mathcal{S}_{\mathcal{G}}, \mathcal{B}_{\mathcal{G}})\}$ **with** $W_{\mathcal{G}} L_{\mathcal{G}} \preceq n^{\frac{d}{2(2\beta_1+d)}}$, **and the discriminator** $\mathcal{F} = \{f : \mathbb{R}^D \to \mathbb{R} \mid f \in \mathcal{NN}(W_{\mathcal{F}}, L_{\mathcal{F}}, \mathcal{S}_{\mathcal{F}}, \mathcal{B}_{\mathcal{F}})\}$ **with** $W_{\mathcal{F}} L_{\mathcal{F}} \preceq n^{\frac{D}{2(2\beta_2+D)}}$. **Suppose that Conditions (C1) - (C2) hold. Let** $\hat{\gamma} = \{\hat{\theta}, \hat{\mathbf{B}}\}$ **be the estimator of** (5), **when** $m > n$, **if** $\lambda_1 = o(1)$, $\lambda_2 = o(\lambda_1)$, $\lambda_3 = o(\lambda_2)$, $(n^{\frac{-\beta_1}{2\beta_1+d}} + n^{\frac{-\beta_2}{2\beta_2+D}}) \log^2 n = o(\lambda_3)$, **we deduce that**

$$\mathbb{P}(\sum_i \mathbb{1}_{\hat{\mathbf{B}}^{[i,:]} \neq \mathbf{0}} = d_0) \to 1, \quad \mathbb{P}(\mathbf{dep}(\hat{\theta}) = l_\theta) \to 1, \quad \mathbb{P}(\|\hat{\theta}\|_0 = n_\theta) \to 1,$$

$$\min_{\theta^* \in \Theta^*} \|\hat{\theta} - \theta^*\|_2 = o_p(1).$$

- **The conditions $\lambda_2 = o(\lambda_1)$ and $\lambda_3 = o(\lambda_1)$ are required because the generator architecture relies on the chosen input dimension.**
- **The condition $\lambda_3 = o(\lambda_2)$ implies that the depth should be identified before determining the sparse structure of the generator architecture.**
- **The requirement that λ_1, λ_2 and λ_3 exceed the statistical error $(n^{\frac{-\beta_1}{2\beta_1+d}} + n^{\frac{-\beta_2}{2\beta_2+D}}) \log^2 n$ is to prevent cases where redundant dimensions or parameters fail to converge to 0 due to randomness.**

Outline

- Background
- Related works on generalization error of GANs
- Theoretical Results
- Method
- Selection consistency
- **Implementation**
- Experiments
- Concluding Remarks

Implementation strategy

- **Dynamic adjustment of λ_1, λ_2, λ_3 during training.**
  - **Start with small penalty parameters, and gradually increase them before subsequently decreasing them.**
- **Sequential selection of λ_1, λ_2 and λ_3 via grid search.**
  - **λ_1 is first chosen such that λ_2 and λ_3 are fixed at 0.**
  - **λ_2 is then selected with λ_1 fixed at the selected value and λ_3 fixed at 0.**
  - **λ_3 is finally determined with λ_1 and λ_2 fixed at the selected values.**
- **Sub-gradient descent algorithm with parameter truncation.**

Algorithm 2: Minibatch stochastic gradient descent training of G-GANs

Input: $\mu_n = \{X_i\}_{i=1}^n$, $\nu_m = \{Z_j\}_{j=1}^m$

Output: $\hat{\theta}$, $\hat{\mathbf{B}}$, $\hat{w}$

Initialization θ, $\mathbf{B}$, w;

for *number of training iterations* **do**

  if *interval* Δ *step and in the first half of training iterations* **then**

    increase the penalty parameters by a factor of δ_1;

  else if *interval* Δ *step* **then**

    decrease the penalty parameters by a factor of δ_2;

  for k *steps* **do**

    Sample mini-batch of b samples $\{Z_1, ..., Z_b\}$ from source distribution ν_m;

    Sample mini-batch of b samples $\{X_1, ..., X_b\}$ from the empirical target distribution μ_n;

    Update the discriminator by ascending its stochastic gradient:
    $\nabla_w \frac{1}{b} \sum_{i=1}^b [f_w(X_i) - f_w(g_\theta(\mathbf{B}Z_i))]$;

  Sample mini-batch of b samples $\{Z_1, ..., Z_b\}$ from source distribution ν_m;

  Update the generator by descending its stochastic gradient:
  $\nabla_{\theta,\mathbf{B}} \frac{1}{b} \sum_{i=1}^b -f_w(g_\theta(\mathbf{B}Z_i)) + \mathcal{L}_{reg}(\mathbf{B}, \theta)$, where
  $\mathcal{L}_{reg}(\mathbf{B}, \theta) = \lambda_1 \|\mathbf{B}\|_{1,2} + \lambda_2 (\sum_{l=1}^{L_\mathcal{G}-1} \|A_l - \mathbf{I}\|_1 + \|c_l\|_1) + \lambda_3 \|\theta\|_1$; **if** *truncating* $\mathbf{B}$ **then**

    Truncating $\mathbf{B}$ according to Algorithm 1;

Return $\hat{\theta}$, $\hat{\mathbf{B}}$, $\hat{w}$.

Outline

- Background
- Related works on generalization error of GANs
- Theoretical Results
- Method
- Selection consistency
- Implementation
- **Experiments**
- Concluding Remarks

Experimental settings

- **The smoothing index $\beta_2 = 1$, inducing the Wasserstein distance and Wasserstein GANs (WGANs).**
- **Two common implementation of WGANs: WGAN-GP and SNGAN. Corresponding G-GANs are G-GAN^W and G-GAN^{SN}.**
- **Two variants of G-GANs:**
  - **G-GANs† with sparsity penalty: $\mathcal{L}_{reg}(\mathbf{B}, \theta) = \lambda_3 \|\theta\|_1$.**
  - **G-GANs‡ with the selection of the input dimension and sparse structure: $\mathcal{L}_{reg}(\mathbf{B}, \theta) = \lambda_1 \|\mathbf{B}\|_{1,2} + \lambda_3 \|\theta\|_1$.**

Evaluation indices

- **Generation quality: Maximum mean discrepancy (MMD) and Fréchet inception distance (FID).**

$$\begin{aligned}\mathbf{MMD}^2(\hat{\mu}_N, \mu_M) &= \|\mathbb{E}_{\hat{X}\sim\hat{\mu}_N}\varphi(\hat{X}) - \mathbb{E}_{X\sim\mu_M}\varphi(X)\|_2^2 \\ &= \frac{1}{N^2}\sum_{i=1}^{N}\sum_{i'=1}^{N} k(\hat{X}_i, \hat{X}_{i'}) + \frac{1}{NM}\sum_{i=1}^{N}\sum_{j=1}^{M} k(\hat{X}_i, X_j) \\ &+ \frac{1}{M^2}\sum_{j=1}^{M}\sum_{j'=1}^{M} k(X_j, X_{j'}),\end{aligned}$$

$$\mathbf{FID}(\hat{\mu}_N, \mu_M) = \|m_{\hat{\mu}_N} - m_{\mu_M}\|_2^2 + Tr(\Sigma_{\hat{\mu}_N} + \Sigma_{\mu_M} - 2(\Sigma_{\hat{\mu}_N}\Sigma_{\mu_M})^{1/2}).$$

- **Input Dimension (Dim.)**
- **The Proportion of zero elements in model parameter $\hat{\theta}$ (Prop.0).**

Numerical simulation

- **Source distribution ν: 10-dimensional standard normal distribution.**
- **Sample size: 10,000 training samples and 2,000 testing samples.**
- **Four generation models:**

(M1) Linear model. Let W be an 100×10 matrix with the elements $10(j-1)$ to $10j$ in the jth column being $[-1, -0.78, -0.56, -0.33, -0.11, 0.11, 0.33, 0.56, 0.78, 1]$, and zero for the remaining components in the jth column. We generate $\mathbf{X}$ by $\mathbf{X} = W\mathbf{Z}$.

(M2) A two-layer rectified linear unit (ReLU) neural network model with sparse connections. Let W_1 be an 50×10 matrix where the elements (i, j) are $[-1, -0.5, 0, 0.5, 1]$ as i varies from $5(j-1)$ to $5j$, and the remaining elements in the j-th column of W_1 are all 0. Let us denote W_2 as an 100×50 matrix where nonzero values appear in rows $2j-1$ to $2j$ of column j, arranged cyclically by using the sequence $[-1, -0.78, -0.56, -0.33, -0.11, 0.11, 0.33, 0.56, 0.78, 1]$. Then, we generate $\mathbf{X}$ by $\mathbf{X} = W_2\sigma(W_1\mathbf{Z})$, where $\sigma(\cdot)$ is the ReLU function.

(M3) Nonlinear model I. The data are generated by $\mathbf{X} = [(\mathbf{Y}[1:20]^T)^2/4, \mathbf{Y}[21:50]^T, exp(\mathbf{Y}[51:70]^T), sin(\mathbf{Y}[71:100]^T \times 20)]^T$, where $\mathbf{Y} = W\mathbf{Z}$ and W are defined in (M1).

(M4) nonlinear model II. The setting is similar to that of M3 except that we generated the data by $\mathbf{X} = [\sqrt{|\mathbf{Y}[1:20]^T|} - 0.1, \mathbf{Y}[21:50]^T, \log(\mathbf{Y}[51:70]^T) + 0.5, \cos(\mathbf{Y}[71:100]^T \times 20)]^T$.

Table 2: The mean of maximum mean discrepancy (MMD), input dimension (Dim.) and proportion of zero elements in model parameters $\hat{\theta}$ (Prop.0) and the corresponding standard deviations (reported in parentheses) for (M1)-(M4). The reported MMD values have been scaled by a factor of 0.0001. The smallest MMDs, lowest input dimensions and highest portion of zero elements are highlighted in bold font.

Dataset	(M1)			(M2)		
Method	MMD(SD)	Dim.(SD)	Prop.0(SD)	MMD(SD)	Dim.(SD)	Prop.0(SD)
WGAN-GP	0.25(0.04)	50.00(0.00)	1.41(0.07)	4.20(4.40)	50.00(0.00)	1.79(0.13)
G-GAN^{W†}	0.60(0.05)	50.00(0.00)	13.75(0.37)	5.10(0.69)	50.00(0.00)	19.51(1.10)
G-GAN^{W‡}	0.34(0.06)	11.80(6.10)	24.88(0.43)	4.86(1.20)	10.40(4.19)	32.35(2.32)
G-GAN^{W}(prop.)	**0.11(0.02)**	**11.80(6.10)**	**91.06(0.10)**	**0.25(0.05)**	**10.40(4.19)**	**93.51(0.18)**
SNGAN	1.03(0.15)	50.00(0.00)	1.50(0.03)	2.77(0.35)	50.00(0.00)	1.60(0.09)
G-GAN^{SN†}	1.07(0.17)	50.00(0.00)	16.60(0.75)	2.68(0.34)	50.00(0.00)	18.50(1.48)
G-GAN^{SN‡}	1.18(0.13)	11.00(4.20)	26.80(0.72)	2.67(0.46)	10.60(4.61)	24.30(4.32)
G-GAN^{SN}(prop.)	**0.14(0.01)**	**11.00(4.20)**	**92.16(0.33)**	**0.55(0.04)**	**10.60(4.61)**	**93.29(0.16)**
Dataset	(M3)			(M4)		
Method	MMD(SD)	Dim.(SD)	Prop.0(SD)	MMD(SD)	Dim.(SD)	Prop.0(SD)
WGAN-GP	3.57(1.27)	50.00(0.00)	1.68(0.12)	1.58(0.44)	50.00(0.00)	1.50(0.07)
G-GAN^{W†}	3.27(1.47)	50.00(0.00)	17.37(1.31)	1.89(0.34)	50.00(0.00)	15.88(0.74)
G-GAN^{W‡}	3.16(0.57)	9.80(4.80)	31.81(1.07)	1.56(0.23)	12.10(5.92)	28.64(0.74)
G-GAN^{W}(prop.)	**0.38(0.07)**	**9.80(4.80)**	**90.63(0.23)**	**0.71(0.05)**	**12.10(5.92)**	**88.76(1.28)**
SNGAN	3.01(1.48)	50.00(0.00)	1.55(0.07)	2.01(0.75)	50.00(0.00)	1.54(0.08)
G-GAN^{SN†}	2.87(1.12)	50.00(0.00)	16.90(0.84)	1.83(0.68)	50.00(0.00)	15.80(0.90)
G-GAN^{SN‡}	2.74(0.64)	11.40(5.22)	27.50(1.34)	1.75(0.37)	8.30(2.93)	26.10(2.31)
G-GAN^{SN}(prop.)	**0.25(0.01)**	**11.40(5.22)**	**91.99(0.83)**	**0.49(0.02)**	**8.30(2.93)**	**92.1(0.81)**

CT slice dataset

- **53,500 CT images obtained from 74 different patients, encompassing 43 males and 31 females.**
- **Total 384 features. 240 features delineate the location of bone structures. 144 features the location of air inclusions within the body.**
- **Training set comprising 50,000 samples and testing set of 3,500 samples.**

Table 3: The mean of maximum mean discrepancy (MMD), input dimension (Dim.) and proportion of zero elements in model parameters $\hat{\theta}$ (Prop.0) and the corresponding standard deviations (reported in parentheses) for CT slices dataset. The reported MMD values have been scaled by a factor of 0.0001. The architecture is identified by the initial input dimension and generator architecture, designated as $d - l \times w$, where d, l, w refer to the initial input dimension, depth and width of the generator, respectively. The smallest MMDs, lowest input dimensions and highest portion of zero elements are highlighted in bold font.

Dataset	CT slices					
Architecture	$64 - 4 \times 256$			$96 - 5 \times 320$		
Method	MMD(SD)	Dim.(SD)	Prop.0(SD)	MMD(SD)	Dim.(SD)	Prop.0(SD)
WGAN-GP	8.83(0.36)	64.00(0.00)	1.90(0.04)	10.95(1.52)	96.00(0.00)	2.12(0.02)
G-GAN^{W}	**8.14(0.04)**	**5.00(1.63)**	**59.30(0.07)**	**8.04(0.02)**	**5.40(1.02)**	**73.14(0.04)**
SNGAN	26.70(10.90)	64.00(0.00)	2.25(0.17)	30.65(11.09)	96.00(0.00)	2.24(0.04)
G-GAN^{SN}	**7.95(0.01)**	**5.30(2.49)**	**57.99(0.35)**	**7.98(0.01)**	**4.60(0.50)**	**73.15(0.03)**

MNIST and FashionMNIST

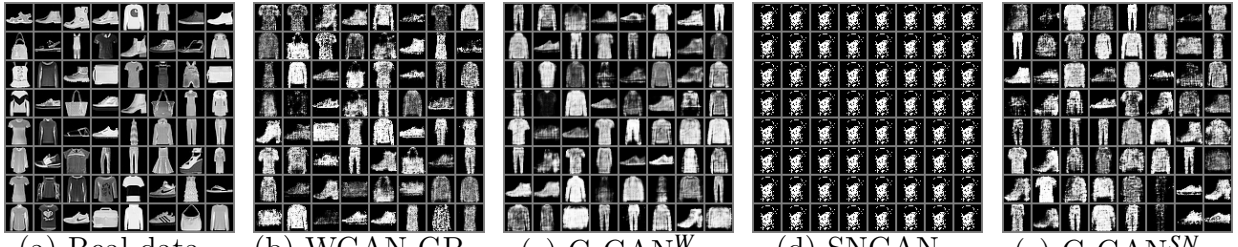
- **The MNIST and FashionMNIST training datasets consist of 60,000 images, each stored in a 28×28 pixel matrix with pixel intensities ranging from 0 to 1.**
- **To satisfy the input requirement of the FNNs, we flatten each 28×28 image pixel matrix into a 784-dimensional vector.**
- **FID is calculated based on 60, 000 generated and training samples.**

Table 4: The mean of Fréchet inception distance (FID), input dimension (Dim.) and proportion of zero elements in model parameters $\hat{\theta}$ (Prop.0) and the corresponding standard deviations (reported in parentheses) for MNIST and FashionMNIST. The smallest FIDs, lowest input dimensions and highest portion of zero elements are highlighted in bold font.

Dataset	MNIST			FashionMNIST		
Method	FID(SD)	Dim.(SD)	Prop.0(SD)	FID(SD)	Dim.(SD)	Prop.0(SD)
WGAN-GP	144.10(51.36)	64.00(0.00)	1.31(0.27)	113.50(3.51)	64.00(0.00)	1.24(0.02)
G-GAN^{W}	**99.86(3.94)**	**7.70(0.47)**	**32.62(0.03)**	**78.65(2.01)**	**7.70(0.47)**	**33.03(0.03)**
SNGAN	239.26(51.14)	64.00(0.00)	3.19(1.06)	340.00(12.77)	64.00(0.00)	3.10(0.05)
G-GAN^{SN}	**105.70(1.45)**	**7.70(0.47)**	**31.97(0.14)**	**125.19(2.93)**	**7.70(2.36)**	**33.23(0.08)**

(a) Real data (b) WGAN-GP (c) G-GAN^{W} (d) SNGAN (e) G-GAN^{SN}

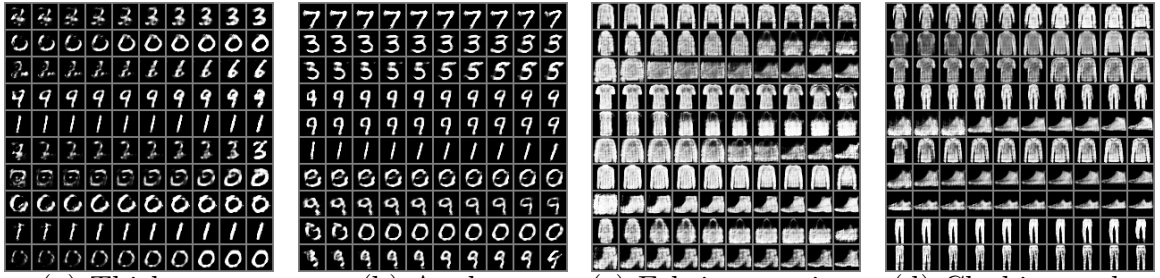
FIGURE 4. Observed images (a) and generated images (b) – (e) by WGAN-GP, G-GAN^{W}, SNGAN and G-GAN^{SN}, respectively for MNIST.



(a) Real data (b) WGAN-GP (c) G-GAN^{W} (d) SNGAN (e) G-GAN^{SN}

FIGURE 5. Observed images (a) and generated images (b)–(e) by WGAN-GP, G-GAN^{W}, SNGAN and G-GAN^{SN}, respectively, for FashionMNIST.

Interpretability



(a) Thickness (b) Angle (c) Fabric quantity (d) Clothing style

FIGURE 6. The manipulation of input variables in the MNIST and FashionMNIST datasets.

- **MNIST: Thickness and angle of inclination of the digits.**
- **FashionMNIST: Fabric quantity and clothing style.**

Outline

- Background
- Related works on generalization error of GANs
- Theoretical Results
- Method
- Selection consistency
- Implementation
- Experiments
- **Concluding Remarks**

We investigate how the input dimension impacts the generalization error of GANs.

- **We first explore the trade-off between the generator approximation and the statistical errors, confirming the existence of an OID that minimizes the generalization error of the GANs.**
- **We propose an adaptable estimation for the input dimension and the corresponding generator architecture.**
- **Rigorous theoretical evidence supports the consistency of the proposed method in terms of both the input dimension and generator architecture.**

Thank you!

Arjovsky, M. and Bottou, L. (2017). Towards principled methods for training generative adversarial networks. In *International Conference on Learning Representations*.

Bailey, B. and Telgarsky, M. J. (2018). Size-noise tradeoffs in generative networks. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pages 6490–6500.

Chen, M., Liao, W., Zha, H., and Zhao, T. (2020). Distribution approximation and statistical estimation guarantees of generative adversarial networks. *arXiv preprint arXiv:2002.03938*.

Dahal, B., Havrilla, A., Chen, M., Zhao, T., and Liao, W. (2022). On deep generative models for approximation and estimation of distributions on manifolds. *Advances in Neural Information Processing Systems*, 35:10615–10628.

Huang, J., Jiao, Y., Li, Z., Liu, S., Wang, Y., and Yang, Y. (2022). An error analysis of generative adversarial networks for learning distributions. *Journal of Machine Learning Research*, 23(116):1–43.

Jiao, Y., Shen, G., Lin, Y., and Huang, J. (2023). Deep nonparametric regression on approximate manifolds: Nonasymptotic error bounds with polynomial prefactors. *The Annals of Statistics*, 51(2):691–716.

Miyato, T., Kataoka, T., Koyama, M., and Yoshida, Y. (2018). Spectral normalization for generative adversarial networks. In *International Conference on Learning Representations*.

Müller, A. (1997). Integral probability metrics and their generating classes of functions. *Advances in Applied Probability*, 29(2):429–443.

Perekrestenko, D., Müller, S., and Bölcskei, H. (2020). Constructive universal high-dimensional distribution generation through deep relu networks. In *International Conference on Machine Learning*, pages 7610–7619. PMLR.

Schreuder, N., Brunel, V.-E., and Dalalyan, A. (2021). Statistical guarantees for generative models without domination. In *Algorithmic Learning Theory*, pages 1051–1071. PMLR.

Yang, Y., Li, Z., and Wang, Y. (2022). On the capacity of deep generative networks for approximating distributions. *Neural Networks*, 145:144–154.