Towards a Better Global Loss Landscape of GANs

Ruoyu Sun, Tiantian Fang, Alex Schwing
University of Illinois at Urbana-Champaign
ruoyus,tf6,aschwing@illinois.edu

Abstract

Understanding of GAN training is still very limited. One major challenge is its non-convex-non-concave min-max objective, which may lead to sub-optimal local minima. In this work, we perform a global landscape analysis of the empirical loss of GANs. We prove that a class of separable-GAN, including the original JS-GAN, has exponentially many bad basins which are perceived as mode-collapse. We also study the relativistic pairing GAN (RpGAN) loss which couples the generated samples and the true samples. We prove that RpGAN has no bad basins. Experiments on synthetic data show that the predicted bad basin can indeed appear in training. We also perform experiments to support our theory that RpGAN has a better landscape than separable-GAN. For instance, we empirically show that RpGAN performs better than separable-GAN with relatively narrow neural nets. The code is available at https://github.com/AilsaF/RS-GAN.

1 Introduction

Generative Adversarial Nets (GANs) are a successful method for learning data distributions. Current theoretical efforts to advance understanding of GANs often focus on statistics or optimization. On the statistics side, Goodfellow et al. built a link between the min-max formulation and the J-S (Jenson-Shannon) distance. Arjovsky and Bottou and Arjovsky et al. proposed an alternative loss function based on the Wasserstein distance. Arora et al. studied the generalization error and showed that both the Wasserstein distance and J-S distance are not generalizable (i.e., both require an exponential number of samples). Nevertheless, Arora et al. argue that the real metric used in practice differs from the two statistical distances, and can be generalizable with a proper discriminator. Bai et al. and Lin et al. analyzed the potential “lack of diversity”: two different distributions can have the same loss, which may cause mode collapse. Bai et al. argue that proper balancing of generator and discriminator permits both generalization and diversity.

On the optimization side, cyclic behavior (non-convergence) is well recognized. This is a generic issue for min-max optimization: a first-order algorithm may cycle around a stable point, converge very slowly or even diverge. The convergence issue can be alleviated by more advanced optimization algorithms such as optimism (Daskalakis et al.), averaging (Yazici et al.) and extrapolation (Gidel et al.).

Besides convergence, another general optimization challenge is to avoid sub-optimal local minima. It is an important issue in non-convex optimization (e.g., Zhang et al., Sun), and has received great attention in matrix factorization and supervised learning. For GANs, the aforementioned works either analyze convex-concave games or perform local analysis. Hence they do not touch the global optimization issue of non-convex problems. Mescheder et al. and Feizi et al. prove global convergence only for simple settings where the true data distribution is a single point or a single Gaussian distribution. The global analysis of GANs for a fairly general data distribution is still a rarely touched direction.

The global analysis of GANs is an interesting direction for the following reasons. First, from a theoretical perspective, it is an indispensable piece for a complete theory. To put our work in...
perspective, we compare representative works in supervised learning with works on GANs in Table 1. **Second**, it may help to understand mode collapse. Bai et al. [7] conjectured that a lack of diversity may be caused by optimization issues, albeit convergence analysis works [65, 8, 34, 11] do not link non-convergence to mode collapse. Thus we suspect that mode collapse is at least partially related to sub-optimal local minima, but a formal theory is still lacking. **Third**, it may help to understand the training process of GANs. Even understanding a simple two-cluster experiment is challenging because the loss values of min-max optimization are fluctuating during training. Global analysis can provide an additional lens in demystifying the training process.

Additional related work is reviewed in Appendix A.

**Challenges and our solutions.** While the idea of a global analysis is natural, there are a few obstacles. First, it is hard to follow a common path of supervised learning [38, 47, 2, 92, 27] to prove global convergence of gradient descent for GANs, because the dynamics of non-convex-non-concave games are much more complicated. Therefore, we resort to a **landscape analysis**. Note that our approach resembles an “equilibrium analysis” in game theory. Second, it was not clear which formulation can cure the landscape issue of JS-GAN. Wasserstein GAN (W-GAN) is a candidate, but its landscape is hard to analyze due to the extra constraints. After analyzing the issue of JS-GAN, we realize that the idea of “paring”, which is implicitly used by W-GAN, is enough to cure the issue. This leads us to consider relativistic paring GANs (RpGANs) [41, 42] that couple the true data and generated data. We prove that RpGANs have a better landscape than separable-GANs (generalization of JS-GAN). Third, it was not clear whether the theoretical finding affects practical training. We make a few conjectures based on our landscape theory and design experiments to verify those. Interestingly, the experiments match the conjectures quite well.

**Our contributions.** This work provides a global landscape analysis of the empirical version of GANs. Our contributions are summarized as follows:

- **Does the original JS-GAN have a good landscape, provably?** For JS-GAN [35], we prove that the outer-minimization problem has exponentially many sub-optimal strict local minima. Each strict local minimum corresponds to a mode-collapse situation. We also extend this result to a class of separable-GANs, covering hinge loss and least squares loss.

- **Is there a way to improve the landscape, provably?** We study a class of relativistic paring GANs (RpGANs) [41] that couple the true data and generated data in the loss function. We prove that the outer-minimization problem of RpGAN has no bad strict local minima, improving upon separable-GANs.

- **Does the improved landscape lead to any empirical benefit?** Based on our theory, we predict thatRpGANs are more robust to data, network width and initialization than their separable counter-parts, and our experiments support our prediction. Although the empirical benefit of RpGANs was observed before [41], the aspects we demonstrate are closely related to our landscape theory. In addition, using synthetic experiments we explain why mode-collapse (as bad basins) can slow down JS-GAN training.

## 2 Difference of Population Loss and Empirical Loss

Goodfellow et al. [35] proved that the population loss of GANs is convex in the space of probability densities. We highlight that this convexity highly depends on a simple property of the population loss, which may vanish in an empirical setting.

Suppose \( p_{\text{data}} \) is the data distribution, \( p_{\theta} \) is a generated distribution and \( D \in C([0,1], \mathbb{R}^d) \), where \( C([0,1], \mathbb{R}^d) \) is the set of continuous functions with domain \( \mathbb{R}^d \) and codomain \((0,1)\). Consider the

---

1In fact, we proposed this loss in a first version of this paper, but later found that [41, 42] considered the same loss. We adopt their name RpGAN from [42].
We focus on the empirical loss in this work. Suppose there are $n$ samples
Appendix A for more.

After the generator update, $y$ (especially with a limited number of samples). See Appendix G for more discussions.

The intuition above is the starting point of this work. We

The proof utilizes two facts: first, the supremum of (infinitely many) convex functions is convex;

second, $\mathbb{E}_{x \sim p_{data}, y \sim p_g} [\log(D(x)) + \log(1 - D(y))]$ is a linear function of $p_g$. The second fact is the

essence of the argument, which we restate below in a more general form.

Claim 2.2 implies that $\min_{p_g} \mathbb{E}_{y \sim p_g} [f_{arb}(y)]$ is a linear function of $p_g$, where $f_{arb}(y)$ is an arbitrary function of $y$.

Claim 2.2 implies that $\min_{p_g} \mathbb{E}_{y \sim p_g} [f_{arb}(y)]$ is a convex problem. One approach to solve it is to
draw finitely many samples (particles) $y_i, i = 1, \ldots, n$ from $p_g$, and approximate the population loss by the empirical loss. See Fig. 1 for a comparison of the probability space and the particle space. For an arbitrarily complicated function such as $f_{arb}(y) = \sin(||y||^8 + 2||y||^3 + \log(||y||^4 + 1))$, the population loss is convex in $p_g$, but clearly the empirical loss is non-convex in $(y_1, \ldots, y_n)$. This example indicates that studying the empirical loss may better reveal the difficulty of the problem (especially with a limited number of samples). See Appendix G for more discussions.

We focus on the empirical loss in this work. Suppose there are $n$ data points $x_1, \ldots, x_n$. We sample $n$ latent variables $z_1, \ldots, z_n \in \mathbb{R}^d$ according to a rule (e.g., i.i.d. Gaussian) and generate artificial data $y_i = G(z_i), i = 1, \ldots, n$. The empirical version of JS-GAN addresses $\min_y \phi_{JS}(Y, X)$ where

$$\phi_{JS}(Y, X) \triangleq \sup_{D} \frac{1}{2n} \sum_{i=1}^{n} [\log(D(x_i)) + \log(1 - D(y_i))].$$

Note that the empirical loss is considered in Arora et al. [5] as well, but they study the generalization properties. We focus on the optimization properties, which is complementary to their work.

3 Landscape Analysis of GANs: Intuition and Toy Results

In this section, we discuss the main intuition and present results for a 2-point distribution.

Intuition of Bad “Local Minima” and Separable-GAN:
Consider an empirical data distribution consisting of two samples $x_1, x_2 \in \mathbb{R}$. The generator produces two data points $y_1, y_2$ to match $x_1, x_2$. We illustrate the training process of JS-GAN in Fig. 2. Initially, $y_1, y_2$ are far from $x_1, x_2$, thus the discriminator can easily separate true data and fake data. After the generator update, $y_1, y_2$ cross the decision boundary to fool the discriminator. Then, after the discriminator update, the decision boundary moves and can again separate true data and fake data. As iterations progress, $y_1, y_2$ and the decision boundary may stay close to $x_1$, causing mode-collapse.

The intuition above is the starting point of this work. We notice that Unterthiner et al. [83], Li and Malik [53] presented somewhat similar intuition, and Kodali et al. [45] suggested the connection between mode collapse and a bad equilibrium. Nevertheless, Li and Malik [53], Kodali et al. [45] do not present a theoretical result, and Unterthiner et al. [83] uses a significantly different formulation from standard GANs. See Appendix A for more.

Figure 1: (a) Population loss: probability density changes; (b) Empirical loss: samples move.

Figure 2: Issue of separable-GAN (including JS-GAN). After updating $G$, fake data crosses boundary to fool $D$; after updating $D$, they are separated by $D$. Fake data may be stuck near $x_1$.
We point out that a major reason for the above issue is a single decision boundary which judges the generated samples. Therefore, this issue exists not only for the JS-GAN, but also for a large class of GANs which we call separable-GANs:

$$\min_Y \sup_f \sum^n_{i=1} h_1(f(x_i)) + h_2(-f(y_i)),$$

where \(h_1, h_2\) are fixed scalar functions, such as \(h_1(t) = h_2(t) = -\log(1 + e^{-t})\) and \(h_1(t) = h_2(t) = -\max\{0, 1 − t\}\), and \(f\) is chosen from a function space (e.g., a set of neural-net functions).

**Pairing as Solution: Rp-GAN.** A natural solution is to use a different “decision boundary” for every generated point, e.g., pairing \(x_i\) and \(y_i\), as illustrated in Fig. 5.

A suitable loss is the relativistic paring GAN (RpGAN)\(^3\)

$$\min_Y \sup_f \sum^n_{i=1} \tilde{h}(f(x_i) − f(y_i)),$$

where \(\tilde{h}\) is a fixed scalar function and \(f\) is chosen from a function space. RS-GAN (relative standard GAN) is a special case where \(h(t) = −\log(1 + e^{-t})\). More specifically, RS-GAN addresses

$$\min_Y \phi_{RS}(Y, X) = \sup_f \frac{1}{n} \sum^n_{i=1} \log \left(1 + \exp(\tilde{f}(y_i) − \tilde{f}(x_i)) \right).$$

W-GAN \(^4\) can be viewed as a variant of RpGAN where \(h(t) = t\), with extra Lipschitz constraint.

We wonder how the issue of separable-GANs relates to “local minima” and how “pairing” helps. We present results for JS-GAN and RS-GAN for the two-point case below.

**Global Landscape of 2-Point Case:** Depending on the positions of \(y_1, y_2\), there are four states \(s_0, s_{1a}, s_{1b}, s_2\). They represent the four cases \(|\{x_1, x_2\} \cap \{y_1, y_2\}| = 0\), \(y_1 = y_2 \in \{x_1, x_2\}\), \(|\{x_1, x_2\} \cap \{y_1, y_2\}| = 1\), and \(|\{x_1, x_2\} \cap \{y_1, y_2\}| = 2\) respectively. Training often starts from the “no-recovery” state \(s_0\), and ideally should end at the “perfect-recovery” state \(s_2\). There are two intermediate states: \(s_{1a}\) means all generated points fall into one mode (“mode collapse”); \(s_{1b}\) means one generated point is the true data point while the other is not a desired data point, which we call “mode dropping”\(^4\). The first three states can transit to each other (assuming continuous change of \(Y\)), but only \(s_{1a}\) can transit to \(s_2\). We illustrate the landscape of \(\phi_{JS}(Y, X)\) and \(\phi_{RS}(Y, X)\) in Fig. 4 by indicating the values in different states. The detailed computation is given next.

**JS-GAN 2-Point Case:** The range of \(\phi_{JS}(Y, X)\) is \([- \log 2, 0]\). The value for the four states are:

**Claim 3.1.** The minimal value of \(\phi_{JS}(Y, X)\) is \(−\log 2\), achieved at \(\{y_1, y_2\} = \{x_1, x_2\}\).

$$\phi_{JS}(Y, X) = \begin{cases} −\log 2 \approx −0.6931 & \text{if } \{x_1, x_2\} = \{y_1, y_2\}, \\ −\log 2/2 \approx −0.3467 & \text{if } |\{x_1, x_2\} \cap \{y_1, y_2\}| = 1, \\ 1/3(2 \log 2 − 3 \log 3) \approx −0.4774 & \text{if } y_1 = y_2 \in \{x_1, x_2\}, \\ 0 & \text{if } |\{x_1, x_2\} \cap \{y_1, y_2\}| = 0. \end{cases}$$

We illustrate the landscape of \(\phi_{JS}(Y, X)\) in Fig. 4a). As a corollary of the above claim, the outer optimization of the original GAN has a bad strict local minimum at state \(s_{1a}\) (a mode-collapse).

**Corollary 3.1.** \(Y = (x_1, x_1)\) is a sub-optimal strict local-min of the function \(g(Y) = \phi_{JS}(Y, X)\).

**RS-GAN 2-Point Case:** The range is still \(\phi_{RS}(Y, X) \in [− \log 2, 0]\). The values are:

**Claim 3.2.** The minimal value of \(\phi_{RS}(Y, X)\) is \(−\log 2\), achieved at \(\{y_1, y_2\} = \{x_1, x_2\}\). In addition,

$$\phi_{RS}(Y, X) = \begin{cases} −\log 2 \approx −0.6931 & \text{if } \{x_1, x_2\} = \{y_1, y_2\}, \\ −\frac{1}{2} \log 2 \approx −0.3466 & \text{if } |\{i : \exists j, \text{ s.t. } x_i = y_j\}|, \\ 0 & \text{otherwise.} \end{cases}$$

\(^3\)Our motivation of considering RpGAN because it breaks locality, thus possibly admitting a better landscape.
\(^4\)This motivation is somewhat different from Jolicoeur-Martineau \(^{41,42}\).
\(^2\)Both may be called mode collapse. Here we differentiate “mode collapse” and “mode dropping”.

Figure 3: Idea of RpGAN: breaking locality by “personalized” judgement.
We present our main theoretical results, extending the landscape results from Assumption 4.1.

The proofs of Claim 3.1 and Claim 3.2 are given in Appendix H. We briefly explain the main insight provided by these proofs. For the mode-collapsed pattern $s_{1a}$, the landscape should be two-dimensional, but we illustrate them in 1D space. Nevertheless, it is still useful for understanding GAN training, as discussed later in Section 5 and Appendix H.

We illustrate $\phi_{RS}(Y, X)$ in Fig. 4(b). Importantly, note that the only basin is the global minimum. In contrast, the landscape of JS-GAN contains a bad basin at a mode-collapsed pattern.

The separable-GAN (SepGAN) problem (empirical loss, function space) is defined in Eq. (5). We have: (i) The global minimal value is $-\frac{1}{2} \min_{i \in [n]} \left[ \log(1 + e^{-t_i}) + \log(1 + e^{t_i}) \right] = -\frac{1}{2} \log 2$, which is of the form $-\frac{1}{2} \log 2$. In contrast, for pattern $s_{1a}$, the loss value of RS-GAN is $-\frac{1}{2} \min_{i \in [n]} \left[ \log(1 + e^{-t_i}) + \log(1 + e^{t_i}) \right] = -\frac{1}{2} \log 2$. Therefore, for the 2-point case, RS-GAN has a better landscape because it avoids the “irregular” value of JS-GAN due to its “pairing”. This insight is the foundation of the general theory presented in the next section.

4 Main Theoretical Results

4.1 Landscape Results in Function Space

We present our main theoretical results, extending the landscape results from $n = 2$ to general $n$.

Denote $\xi(m) \triangleq \sup_{t \in \mathbb{R}} \left( h_1(t) + mh_2(t) \right)$.

Assumption 4.1. $\sup_{t \in \mathbb{R}} h_1(t) = \sup_{t \in \mathbb{R}} h_2(t) = 0$.

Assumption 4.2. $\xi(m) > m\xi(1), \forall m \in [2, n]$.

Assumption 4.3. $\xi(m) < \xi(m - 1), \forall m \in [1, n]$.

It is easy to prove that under Assumption 4.1, $\xi(m - 1) \geq \xi(m) \geq m\xi(1)$ always holds. Assumption 4.2 and Assumption 4.3 require strict inequalities, thus do not always hold (e.g., for constant functions). Nevertheless, most non-constant functions satisfy these assumptions.

The separable-GAN (SepGAN) problem (empirical loss, function space) is

$$\min_{Y \in \mathbb{R}^{d \times n}} g_{SP}(Y), \text{ where } g_{SP}(Y) = \frac{1}{2n} \sup_{f \in C(\mathbb{R}^d)} \sum_{i=1}^{n} \left[ h_1(f(x_i)) + h_2(-f(y_i)) \right].$$

Theorem 1. Suppose $x_1, x_2, \ldots, x_n \in \mathbb{R}^d$ are distinct. Suppose $h_1, h_2$ satisfy Assumptions 4.1 and 4.2. Then for separable-GAN loss $g_{SP}(Y)$ defined in Eq. (5), we have: (i) The global minimal value is $-\frac{1}{2} \sup_{t \in \mathbb{R}} \left( h_1(t) + h_2(-t) \right)$, which is achieved iff $\{y_1, \ldots, y_n\} = \{x_1, \ldots, x_n\}$. (ii) If $y_i \in \{x_1, \ldots, x_n\}, i \in \{1, 2, \ldots, n\}$ and $y_i = y_j$ for some $i \neq j$, then $Y$ is a sub-optimal strict local minimum. Therefore, $g_{SP}(Y)$ has $(n^n - n!)$ sub-optimal strict local minima.

Remark 1: $h_1(t) = h_2(t) = -\log(1 + e^{-t})$ satisfy Assumptions 4.1 and 4.2, thus Theorem 1 applies to JS-GAN. It also applies to hinge-GAN with $h_1(t) = h_2(t) = -\max\{0, 1 - t\}$ and LS-GAN (least-square GAN) with $h_1(t) = -(1 - t)^2, h_2(t) = -t^2$.

Next we consider RpGANs. The RpGAN problem (empirical loss, function space) is

$$\min_{Y \in \mathbb{R}^{d \times n}} g_{R}(Y), \text{ where } g_{R}(Y) = \frac{1}{n} \sup_{f \in C(\mathbb{R}^d)} \sum_{i=1}^{n} \left[ h(f(x_i) - f(y_i)) \right].$$

Definition 4.1. (global-min-reachable) We say a point $w$ is global-min-reachable for a function $F(w)$ if there exists a continuous path from $w$ to one global minimum of $F$ along which the value of $F(w)$ is non-increasing.
Assumption 4.4. $\sup_{t \in \mathbb{R}} h(t) = 0$ and $h(0) < 0$.

Assumption 4.5. $h$ is a concave function in $\mathbb{R}$.

Theorem 2. Suppose $x_1, x_2, \ldots, x_n \in \mathbb{R}^d$ are distinct. Suppose $h$ satisfies Assumptions 4.4 and 4.5 then for RpGAN loss $g_h$ defined in Eq. (8): (i) The global minimal value is $h(0)$, which is achieved iff $\{y_1, \ldots, y_n\} = \{x_1, \ldots, x_n\}$. (ii) Any $Y$ is global-min-reachable for the function $g_h(Y)$.

This result sanity checks the loss $g_h(Y)$: its global minimizer is indeed the desired empirical distribution. In addition, it establishes a significantly different optimization landscape for RpGAN.

Remark 1: The existence of a decreasing path does not necessarily mean an algorithm can follow it. Nevertheless, our results already distinguish SepGAN and RpGAN. We will illustrate that these results can improve our understanding of GAN training in Sec. 5 and present experiments supporting our theory in Sec. 6.

Remark 2: The two results rely on a few assumptions of neural-nets including Assumption 4.6. These assumptions can be satisfied by certain over-parameterized neural-nets, in which case $W$ is a certain dense subset of $\mathbb{R}^K$ or $\mathbb{R}^K$ itself. For details see Appendix K.1

4.2 Landscape Results in Parameter Space

We now consider a deep net generator $G_w$ with $w \in \mathbb{R}^K$ and a deep net discriminator $f_\theta$ with $\theta \in \mathbb{R}^d$. Different from before, where we optimize over $y_i$ and $f$ (function space), we now optimize over $w$ and $\theta$ (parameter space).

We first present a technical assumption. For $Z = (z_1, \ldots, z_n) \in \mathbb{R}^{d_z \times n}$, $Y = (y_1, \ldots, y_n) \in \mathbb{R}^{d_y \times n}$ and $W \subseteq \mathbb{R}^K$, define a set $G^{-1}(Y; Z, W) \triangleq \{w \in W \mid G_w(z_1) = y_1, \forall i\}$.

Assumption 4.6. (path-keeping property of generator net): For any distinct $z_1, \ldots, z_n \in \mathbb{R}^{d_z}$, any continuous path $Y(t), t \in [0, 1]$ in the space $\mathbb{R}^{d_y \times n}$ and any $w_0 \in G^{-1}(Y(0); Z, W)$, there is continuous path $w(t), t \in [0, 1]$ such that $w(0) = w_0$ and $Y(t) = G_{w(t)}(Z)$, $t \in [0, 1]$.

Intuitively, this assumption relates the paths in the function space to the paths in the parameter space, thus the results in function space can be transferred to the results in parameter space. The formal results involve two extra assumptions on representation power of $f_\theta$ and $G_w$ (see Appendix K for details). Informal results are as follows:

Proposition 1. (informal) Consider the separable-GAN problem $\min_{w \in \mathbb{R}^K} \varphi_{\text{sep}}(w)$, where

$$\varphi_{\text{sep}}(w) = \sup_{\theta} \frac{1}{2n} \sum_{i=1}^{n} [h_1(f_\theta(x_i)) + h_2(-f_\theta(G_w(z_i)))].$$

(7)

Suppose $h_1, h_2$ satisfy the assumptions of Theorem 7. Suppose $G_w$ satisfies Assumption 4.6 (with certain $W$). Suppose $f_\theta$ and $G_w$ have enough representation power (formalized in Appendix K).

Then there exist at least $(n^n - n!)$ distinct $w \in W$ that are not global-min-reachable for $\varphi_{\text{sep}}(w)$.

Proposition 2. (informal) Consider the RpGAN problem $\min_{w \in \mathbb{R}^K} \varphi_R(w)$, where

$$\varphi_R(w) = \sup_{\theta} \frac{1}{n} \sum_{i=1}^{n} [h(f_\theta(x_i)) - f_\theta(G_w(z_i))].$$

(8)

Suppose $h$ satisfies the assumptions of Theorem 2. Suppose $G_w$ and $f_\theta$ satisfy the same assumptions as Proposition 7. Then any $w \in W$ is global-min-reachable for $\varphi_R(w)$.

Remark 1: The existence of a decreasing path does not necessarily mean an algorithm can follow it. Nevertheless, our results already distinguish SepGAN and RpGAN. We will illustrate that these results can improve our understanding of GAN training in Sec. 5 and present experiments supporting our theory in Sec. 6.

Remark 2: The two results rely on a few assumptions of neural-nets including Assumption 4.6. These assumptions can be satisfied by certain over-parameterized neural-nets, in which case $W$ is a certain dense subset of $\mathbb{R}^K$ or $\mathbb{R}^K$ itself. For details see Appendix K.1
We discuss a two-cluster experiment, an extension of two-point generation, in order to build a link between theory and practice. We proved that a mode collapse might be far away from the basins at \( Y^* \) (e.g., from iteration \( 2000 \) to \( 2500 \)). In short, the escape for JS-GAN.

These results distinguish the SepGAN and RpGAN landscapes. Theoretically, there is evidence that adding neurons can smooth the landscape of deep nets (e.g., eliminating bad basins in neural-nets [50]), thus wide nets might help escape basins in the \((Y, D)\)-space faster. In short, the effect of bad basins may be mitigated via the following factors: (i) proper initial \( D \) and \( Y \); (ii) long enough training time; (iii) wide neural-nets; (iv) enough hyper-parameter tuning. These factors make it relatively hard to detect the existence of bad basins and their influences. We support our landscape theory, by identifying differences of SepGAN and RpGAN in synthetic and real-data experiments.

4.3 Discussion of Implications

These results distinguish the SepGAN and RpGAN landscapes. Theoretically, there is evidence regarding the benefit of losses without sub-optimal basins. Bovier et al. [17] proved that it takes the Langevin diffusion at least \( e^{\omega(h)} \) time to escape a depth-\( h \) basin. A recent work [91] proved that the hitting time of SGLD (stochastic gradient Langevin dynamics) is positively related to the height of the barrier, and SGLD may escape basins with low barriers relatively fast. The theoretical insight is that a landscape without a bad basin permits better quality solutions or a faster convergence to good-quality solutions.

We now discuss the possible gap between our theory and practice. We proved that a mode collapse \( Y^* \) is a bad basin in the generator space, which indicates that \((Y^*, D^*(Y^*))\) is an attractor in the joint space of \((Y, D)\) and hard to escape by gradient descent ascent (GDA). In GAN training, the dynamics are not the same as GDA dynamics due to various reasons (e.g., sampling, unequal \( D \) and \( G \) updates), and basins could be escaped with enough training time (e.g., 91). In addition, a randomly initialized \((Y, D)\) might be far away from the basins at \((Y^*, D^*(Y^*))\), and properly chosen hyper-parameters (e.g., learning rate) may re-position the dynamics so as to avoid attraction to bad basins. Further, it is known that adding neurons can smooth the landscape of deep nets (e.g., eliminating bad basins in neural-nets [50]), thus wide nets might help escape basins in the \((Y, D)\)-space faster. In short, the effect of bad basins may be mitigated via the following factors: (i) proper initial \( D \) and \( Y \); (ii) long enough training time; (iii) wide neural-nets; (iv) enough hyper-parameter tuning. These factors make it relatively hard to detect the existence of bad basins and their influences. We support our landscape theory, by identifying differences of SepGAN and RpGAN in synthetic and real-data experiments.

5 Case Study of Two-Cluster Experiments

Although in Section 3 we argue that, intuitively, mode collapse can happen for training JS-GAN for two-point generation, it does not necessarily mean mode collapse really appears in practical training. We discuss a two-cluster experiment, an extension of two-point generation, in order to build a link between theory and practice. We aim to understand the following question: does mode collapse really appear as a “basin”, and how does it affect training?

Suppose the true data are two clusters around \( c_1 = 0 \) and \( c_2 = 4 \). We sample 100 points from the two clusters as \( x_i \)'s, and sample \( z_1, \ldots, z_{100} \) uniformly from an interval. We use 4-layer neural-nets for the discriminator and generator. We use the non-saturating versions of JS-GAN and RS-GAN.

**Mode collapse as bad basin can appear.** We visualize the movement of fake data in Fig. 5, and plot the loss value of \( D \) (indicating the discriminator) over iterations in Fig. 4(a,b). Interestingly, the minimal \( D \) losses are around 0.48, which is the value of \( \phi_{JS} \) at state \( s_{1a} \). It is easy to check that the optimal \( D = D^*(s_{1a}) \) for a mode collapse state \( s_{1a} \) satisfies \( \{D(c_1), D(c_2)\} = \{1, 1/3\} \), and Fig. 5(c) shows that at iteration 2800 the \( D \) actually becomes \( D^* \). This provides a concrete example that training gets stuck at a mode collapse due to the bad-basin-effect. We also notice that there are a few more attempts to approach the bad attractor \((s_{1a}, D^*(s_{1a}))\) (e.g., from iteration 2000 to 25000). In RS-GAN training, the minimal loss is around 0.35, which is also the value of \( \phi_{RS} \) at state \( s_{1a} \). The attracting power of \((s_{1a}, D^*(s_{1a}))\) is weaker than for JS-GAN. Thus it only attracts the iterates for a very short time. RS-GAN needs 800 iterations to escape, which is about 3 times faster than the escape for JS-GAN.

**Effect of width:** We see a clear effect of width on convergence speed. As the networks become wider, both JS-GAN and RS-GAN converge faster. We find that the reason of faster convergence is because wider nets make JS-GAN escape mode collapse faster. See details in Appendix B.

More experiment details and findings are presented in Appendix B.

![Figure 5: Training process of JS-GAN and RS-GAN for two-cluster data. True data are red, fake data are blue. RS-GAN escapes from mode collapse faster than JS-GAN.](image-url)
Figure 6: (a) and (b); Evolution of $D$ loss over iterations. RS-GAN is 3-4× faster than JS-GAN. (c) For JS-GAN training in (a), we plot $(Y,D)$ together at iteration 2800. $Y$ are represented in blue points, and they are near $c_1 = 0$. $D$ is near the optimal $D^*(s_{14})$ since $D(0) \approx 1/3$ and $D(4) \approx 1$. Interestingly, this bad attractor $(Y,D)$ is similar to the one discussed in Fig. 1, so the intuition of “local-min” is verified in (c).

### Table 2

| Real Dataset             | Inception Score ↑ | FID ↓ | FID Gap | Model size |
|--------------------------|-------------------|-------|---------|------------|
| WGAN-GP                  | 6.68±0.06         | 39.66 | 15.34   | 100%       |
| JS-GAN                   | 6.27±0.10         | 49.13 | 0.91    | 100%       |
| RS-GAN                   | 7.02±0.07         | 33.79 | 15.34   | 100%       |
| JS-GAN+ SN               | 7.42±0.08         | 28.07 | 0.91    | 100%       |
| RS-GAN+ SN               | 7.32±0.08         | 27.16 | 0.91    | 100%       |
| JS-GAN+SN; GD channel/2  | 6.85±0.08         | 33.90 | 15.34   | 100%       |
| RS-GAN+SN; GD channel/2  | 6.74±0.04         | 32.74 | 15.34   | 100%       |
| JS-GAN + SN; GD channel/4| 5.83±0.07         | 52.63 | 0.91    | 100%       |
| RS-GAN + SN; GD channel/4| 5.94±0.09         | 45.37 | 0.91    | 100%       |

| ResNet                   | Inception Score ↑ | FID ↓ | FID Gap | Model size |
|--------------------------|-------------------|-------|---------|------------|
| JS-GAN+ SN               | 8.12±0.14         | 20.13 | 0.82    | 100%       |
| RS-GAN + SN               | 7.92±0.13         | 19.31 | 0.82    | 100%       |
| JS-GAN + SN; GD channel/2| 7.67±0.04         | 23.29 | 1.51    | 27.5%      |
| RS-GAN + SN; GD channel/2| 7.63±0.07         | 21.78 | 1.51    | 27.5%      |
| JS-GAN + SN; GD channel/4| 6.65±0.06         | 45.20 | 13.94   | 10.4%      |
| RS-GAN+ SN; GD channel/4  | 7.08±0.05         | 31.26 | 13.94   | 10.4%      |
| JS-GAN + SN; BottleNeck   | 7.60±0.07         | 26.98 | 1.54    | 16.8%      |
| RS-GAN+SN; BottleNeck     | 7.57±0.09         | 25.44 | 1.54    | 16.8%      |

### 6 Real Data Experiments

RpGANs have been tested by Jolicoeur-Martineau [41], and are shown to be better than their SepGAN counterparts in a variety of settings. In addition, RpGAN and its variants have been used in super-resolution (ESRGAN) [85] and a few recent GANs [87, 13]. Therefore, the effectiveness of RpGANs has been justified to some extent. We do not attempt to re-run the experiments merely for the purpose of justification. Instead, our goal is to use experiments to support our landscape theory.

Based on the discussions in Sec. 2, Sec. 3, and Sec. 5, we conjecture that RpGANs have a bigger advantage over SepGAN (A) with narrow deep nets, (B) in high resolution image generation, (C) with imbalanced data. Finally, (D) there exists some bad initial $D$ that makes SepGANs much worse than RpGANs. In the main text, we present results on the logistic loss (i.e., JS-GAN and RS-GAN). Results on other losses are given in the appendix.

**Experimental setting for (A).** For (A), we test on CIFAR-10 and STL-10 data. For the optimizer, we use Adam with the discriminator’s learning rate 0.0002. For CIFAR-10 on ResNet, we set $\beta_1 = 0$ and $\beta_2 = 0.9$ in Adam; for others, $\beta_1 = 0.5$ and $\beta_2 = 0.999$. We tune the generator’s learning rate and run 100k iterations in total. We report the Inception score (IS) and Frechét Inception distance (FID). IS and FID are evaluated on 50k and 10k samples respectively. More details of the setting are shown in Appendix E.1 and the experimental settings for other cases besides (A) are shown in the corresponding parts in the appendix. Generated images are shown in Appendix F.

**Regular architecture and effect of spectral norm (SN).** We use two neural architectures in [67]: standard CNN and ResNet, and report results in Table 2. First, without spectral normalization (SN), RS-GAN achieves much higher accuracy than JS-GAN and WGAN-GP on CIFAR-10. Second, with

---

4That paper tested a number of variants, and some of them are not directly covered by our results.
SN, RS-GAN achieves 1-2 points lower FID score than JS-GAN, i.e., it’s slightly better. We suspect that SN smoothens the landscape, thus greatly reducing the gap between JS-GAN and RS-GAN. Note that the scores of JS-GAN and WGAN-GP (both without and with SN) are comparable to or better than the scores in Table 2 of Miyato et al. [67].

**Narrow nets.** For both CNN and ResNet, we reduce the number of channels for all convolutional layers in the generator and discriminator to (1) half, (2) quarter and (3) bottleneck (for ResNet structure). The experimental results are provided in Table 2. We consider the gap between RS-GAN and JS-GAN for regular width as a baseline. For narrow nets, the gap between RS-GAN and JS-GAN is similar or larger in most cases, and can be much larger (e.g. > 13 FID) in some cases. The fluctuations in the gaps are consistent with landscape theory: if JS-GAN training gets stuck at a bad basin then the performance is bad; if it converges to a good basin, then the performance is reasonably good. In CIFAR-10, compared to SN-GAN with the conventional ResNet (FID=20.13), we can achieve a relatively close result by using RS-GAN with 28% parameters (half channel, FID=21.78).

**High resolution data experiments.** Sec. 2 discusses that the non-convexity of JS-GAN will become a more severe issue when the number of samples is limited compared to the data space (e.g., high resolution space or limited data points). We conduct experiments with LSUN Church and Tower images of size $256 \times 256$. RS-GAN can generate higher visual quality images than JS-GAN (Appendix E). Similarly, using another model architecture, [41] achieves a better FID score with RSGAN on the CAT dataset, which contains a small number of images (e.g., 2k $256 \times 256$ images).

**Imbalanced data experiments.** For imbalanced data, we find more evidence for the existence of JS-GAN’s bad basins. The reason: JS-GAN would have a deeper bad basin, and hence a higher chance to get stuck. We conduct ablation experiments on 2-cluster data and MNIST. Both cases show that JS-GAN ends up with mode collapse while RS-GAN can generate data with proportions similar to the imbalanced true data. Check Appendix C for more.

**Bad initial point experiments.** A better landscape is more robust to initialization. On MNIST data, we find a discriminator (not random) which permits RS-GAN to converge to a much better solution than JS-GAN when used as the starting point. The FID scores are reported in the table to the right. The gap is at least 30 FID scores (a much higher gap than the gap for a random initialization). Check Appendix C for more.

**Combining with EMA.** It is known that non-convergence can be alleviated via EMA [88], and our theory predicts that the global landscape issue can be alleviated by RpGAN. Non-convergence and global landscape are orthogonal: no matter whether iterates are near a sub-optimal local basin or a globally-optimal basin, the algorithm may cycle. Therefore, we conjecture that the effect of EMA and the effect of RS-GAN are “additive”. Our simulations show that EMA can improve both JS-GAN and RS-GAN, and the gap is approximately preserved after adding EMA. Combining EMA and RS-GAN, we achieve a similar result to the baseline (JS-GAN + SN, no EMA, FID = 20.13) using 16.8% parameters (Resnet with bottleneck plus EMA, FID=21.38). See Appendix E.1 for more.

**General RpGAN:** We conduct additional experiments on other losses, including hinge loss and least squares loss. See Appendix E.2 and E.3 for more.

## 7 Conclusion

Global optimization landscape, together with statistical analysis and convergence analysis, are important theoretical angles. In this work, we study the global landscape of GANs. Our major questions are: (1) Does the original JS-GAN formulation have a good landscape? (2) If not, is there a simple way to improve the landscape in theory? (3) Does the improved landscape lead to better performance? First, studying the empirical versions of SepGAN (extension of JS-GAN) we prove that it has exponentially many bad basins, which are mode-collapse patterns. Second, we prove that a simple coupling idea (resulting in RpGAN) can remove bad basins in theory. Finally, we verify a few predictions based on the landscape theory, e.g., RS-GAN has a bigger advantage over JS-GAN for narrow nets.
Acknowledgements

This work is supported in part by NSF under Grant # 1718221, 2008387, 1755847 and MRI #1725729, and NIFA award 2020-67021-32799. We thank Sewoong Oh for pointing out the connection of the earlier version of our work to [41].

References

[1] J. Adler and S. Lunz. Banach wasserstein gan. In NeurIPS, 2018.
[2] Z. Allen-Zhu, Y. Li, and Z. Song. A convergence theory for deep learning via over-parameterization. In ICML, 2019.
[3] M. Arjovsky and L. Bottou. Towards principled methods for training generative adversarial networks. In ICLR, 2017.
[4] M. Arjovsky, S. Chintala, and L. Bottou. Wasserstein gan. In ICML, 2017.
[5] S. Arora, R. Ge, Y. Liang, T. Ma, and Y. Zhang. Generalization and equilibrium in generative adversarial nets (GANs). In ICML, 2017.
[6] W. Azizian, I. Mitliagkas, S. Lacoste-Julien, and G. Gidel. A tight and unified analysis of extragradient for a whole spectrum of differentiable games. arXiv preprint arXiv:1906.05945, 2019.
[7] Y. Bai, T. Ma, and A. Risteski. Approximability of discriminators implies diversity in gans. arXiv preprint arXiv:1806.10586, 2018.
[8] D. Balduzzi, S. Racaniere, J. Martens, J. Foerster, K. Tuyls, and T. Graepel. The mechanics of n-player differentiable games. arXiv preprint arXiv:1802.05642, 2018.
[9] P. L. Bartlett, D. J. Foster, and M. J. Telgarsky. Spectrally-normalized margin bounds for neural networks. In NeurIPS, 2017.
[10] Y. Bengio and Y. LeCun. Scaling learning algorithms towards AI. In Large Scale Kernel Machines. MIT Press, 2007.
[11] H. Berard, G. Gidel, A. Almahairi, P. Vincent, and S. Lacoste-Julien. A closer look at the optimization landscapes of generative adversarial networks. arXiv preprint arXiv:1906.04848, 2019.
[12] D. Berthelot, T. Schumm, and L. Metz. Began: Boundary equilibrium generative adversarial networks. arXiv preprint arXiv:1703.10717, 2017.
[13] D. Berthelot, P. Milanfar, and I. Goodfellow. Creating high resolution images with a latent adversarial generator. arXiv preprint arXiv:2003.02365, 2020.
[14] S. Bhojanapalli, B. Neyshabur, and N. Srebro. Global optimality of local search for low rank matrix recovery. In NeurIPS, 2016.
[15] M. Bianchini and M. Gori. Optimal learning in artificial neural networks: A review of theoretical results. Neurocomputing, 1996.
[16] M. Bińkowski, D. J. Sutherland, M. Arbel, and A. Gretton. Demystifying mmd gans. In ICLR, 2018.
[17] A. Bovier, M. Eckhoff, V. Gayrard, and M. Klein. Metastability in reversible diffusion processes i. sharp asymptotics for capcities and exit times. JEMS, 2004.
[18] A. Brock, J. Donahue, and K. Simonyan. Large scale gan training for high fidelity natural image synthesis. arXiv preprint arXiv:1809.11096, 2018.
[19] Y. Chi, Y. M. Lu, and Y. Chen. Nonconvex optimization meets low-rank matrix factorization: An overview. IEEE Transactions on Signal Processing, 67(20):5239–5269, 2019.
[20] C. Chu, J. Blanchet, and P. Glynn. Probability functional descent: A unifying perspective on gans, variational inference, and reinforcement learning. arXiv preprint arXiv:1901.10691, 2019.
[21] R. W. A. Cully, H. J. Chang, and Y. Demiris. Magan: Margin adaptation for generative adversarial networks. arXiv preprint arXiv:1704.03817, 2017.
[22] C. Daskalakis and I. Panageas. The limit points of (optimistic) gradient descent in min-max optimization. In NeurIPS, 2018.

[23] C. Daskalakis, A. Ilyas, V. Syrgkanis, and H. Zeng. Training gans with optimism. In ICLR, 2018.

[24] I. Deshpande, Z. Zhang, and A. Schwing. Generative modeling using the sliced wasserstein distance. In CVPR, 2018.

[25] I. Deshpande, Y.-T. Hu, R. Sun, A. Pyrros, N. Siddiqui, S. Koyejo, Z. Zhao, D. Forsyth, and A. G. Schwing. Max-Sliced Wasserstein Distance and its use for GANs. In CVPR, 2019.

[26] T. Ding, D. Li, and R. Sun. Sub-optimal local minima exist for almost all over-parameterized neural networks. arXiv preprint arXiv:1911.01413, 2019.

[27] S. S. Du, J. D. Lee, H. Li, L. Wang, and X. Zhai. Gradient descent finds global minima of deep neural networks. arXiv preprint arXiv:1811.03804, 2018.

[28] F. Farnia and A. Ozdaglar. Gans may have no nash equilibria. arXiv preprint arXiv:2002.09124, 2020.

[29] F. Farnia and D. Tse. A convex duality framework for gans. In NeurIPS, 2018.

[30] S. Feizi, F. Farnia, T. Ginart, and D. Tse. Understanding gans: the lqg setting. arXiv preprint arXiv:1710.10793, 2017.

[31] R. Ge, J. D. Lee, and T. Ma. Matrix completion has no spurious local minimum. In NeurIPS, 2016.

[32] M. Geiger, S. Spigler, S. d’Ascoli, L. Sagun, M. Baity-Jesi, G. Biroli, and M. Wyart. The jamming transition as a paradigm to understand the loss landscape of deep neural networks. arXiv preprint arXiv:1809.09349, 2018.

[33] G. Gidel, H. Berard, G. Vignoud, P. Vincent, and S. Lacoste-Julien. A variational inequality perspective on generative adversarial networks. arXiv preprint arXiv:1802.10551, 2018.

[34] G. Gidel, R. A. Hemmat, M. Pezeshki, R. Leprilol, G. Huang, S. Lacoste-Julien, and I. Mitliagkas. Negative momentum for improved game dynamics. In AISTATS, 2019.

[35] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In NeurIPS, 2014.

[36] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. Courville. Improved training of wasserstein gans. In NeurIPS, 2017.

[37] X. Huang, Y. Li, O. Poursaeed, J. Hopcroft, and S. Belongie. Stacked generative adversarial networks. In CVPR, 2017.

[38] A. Jacot, F. Gabriel, and C. Hongler. Neural tangent kernel: Convergence and generalization in neural networks. In NeurIPS, 2018.

[39] C. Jin, P. Netrapalli, and M. I. Jordan. Minmax optimization: Stable limit points of gradient descent ascent are locally optimal. arXiv preprint arXiv:1902.00618, 2019.

[40] R. Johnson and T. Zhang. A framework of composite functional gradient methods for generative adversarial models. IEEE transactions on pattern analysis and machine intelligence, 2019.

[41] A. Jolicoeur-Martineau. The relativistic discriminator: a key element missing from standard gan. In ICLR, 2018.

[42] A. Jolicoeur-Martineau. On relativistic f-divergences. In ICML, 2019.

[43] T. Karras, S. Laine, and T. Aila. A style-based generator architecture for generative adversarial networks. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 4401–4410, 2019.

[44] T. Karras, S. Laine, M. Aittala, J. Hellsten, J. Lehtinen, and T. Aila. Analyzing and improving the image quality of stylegan. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 8110–8119, 2020.

[45] N. Kodali, J. Abernethy, J. Hays, and Z. Kira. On convergence and stability of gans. arXiv preprint arXiv:1705.07215, 2017.
[46] S. Kolouri, C. E. Martin, and G. K. Rohde. Sliced-wasserstein autoencoder: An embarrassingly simple generative model. arXiv preprint arXiv:1804.01947, 2018.

[47] J. Lee, L. Xiao, S. Schoenholz, Y. Bahri, R. Novak, J. Sohl-Dickstein, and J. Pennington. Wide neural networks of any depth evolve as linear models under gradient descent. In Advances in neural information processing systems, pages 8572–8583, 2019.

[48] Q. Lei, J. D. Lee, A. G. Dimakis, and C. Daskalakis. Sgd learns one-layer networks in wgans. arXiv preprint arXiv:1910.07030, 2019.

[49] C.-L. Li, W.-C. Chang, Y. Cheng, Y. Yang, and B. Póczos. Mmd gan: Towards deeper understanding of moment matching network. In NeurIPS, 2017.

[50] D. Li, T. Ding, and R. Sun. Over-parameterized deep neural networks have no strict local minima for any continuous activations. arXiv preprint arXiv:1812.11039, 2018.

[51] J. Li, A. Madry, J. Peebles, and L. Schmidt. On the limitations of first-order approximation in gan dynamics. arXiv preprint arXiv:1706.09884, 2017.

[52] J. Li, A. Madry, J. Peebles, and L. Schmidt. Towards understanding the dynamics of generative adversarial networks. In ICML, 2018.

[53] K. Li and J. Malik. Implicit maximum likelihood estimation. arXiv preprint arXiv:1809.09087, 2018.

[54] Y. Li, A. G. Schwing, K.-C. Wang, and R. Zemel. Dualing GANs. In NeurIPS, 2017.

[55] S. Liang, R. Sun, Y. Li, and R. Srikant. Understanding the loss surface of neural networks for binary classification. arXiv preprint arXiv:1803.00909, 2018.

[56] S. Liang, R. Sun, R. Srikant. Revisiting landscape analysis in deep neural networks: Eliminating decreasing paths to infinity. arXiv preprint arXiv:1912.13472, 2019.

[57] Z. Lin, A. Khetan, G. Fanti, and S. Oh. Pacgan: The power of two samples in generative adversarial networks. In NeurIPS, 2018.

[58] S. Liu and K. Chaudhuri. The inductive bias of restricted f-gans. arXiv preprint arXiv:1809.04542, 2018.

[59] R. Livni, S. Shalev-Shwartz, and O. Shamir. On the computational efficiency of training neural networks. In NeurIPS, 2014.

[60] A. V. Makkuva, A. Taghvaei, S. Oh, and J. D. Lee. Optimal transport mapping via input convex neural networks. arXiv preprint arXiv:1908.10962, 2019.

[61] X. Mao, Q. Li, H. Xie, R. Y. K. Lau, Z. Wang, and S. P. Smolley. Least Squares Generative Adversarial Networks. arXiv e-prints, 2016.

[62] X. Mao, Q. Li, H. Xie, R. Y. Lau, Z. Wang, and S. Paul Smolley. Least squares generative adversarial networks. In ICCV, 2017.

[63] E. V. Mazumdar, M. I. Jordan, and S. S. Sastry. On finding local nash equilibria (and only local nash equilibria) in zero-sum games. arXiv preprint arXiv:1901.00838, 2019.

[64] L. Mescheder, A. Geiger, and S. Nowozin. Which training methods for gans do actually converge? In ICML, 2018.

[65] L. Metz, B. Poole, D. Pfau, and J. Sohl-Dickstein. Unrolled generative adversarial networks. In ICLR, 2017.

[66] T. Miyato, T. Kataoka, M. Koyama, and Y. Yoshida. Spectral normalization for generative adversarial networks. In ICLR, 2018.

[67] S. Mohamed and B. Lakshminarayanan. Learning in implicit generative models. arXiv preprint arXiv:1610.03483, 2016.

[68] Y. Mroueh and T. Sercu. Fisher gan. In NeurIPS, 2017.

[69] Y. Mroueh, T. Sercu, and V. Goel. Megan: Mean and covariance feature matching gan. arXiv preprint arXiv:1702.08398, 2017.
[71] V. Nagarajan and J. Z. Kolter. Gradient descent gan optimization is locally stable. In NeurIPS, 2017.

[72] Q. Nguyen and M. Hein. The loss surface of deep and wide neural networks. In ICML, 2017.

[73] Q. Nguyen, M. C. Mukkamala, and M. Hein. On the loss landscape of a class of deep neural networks with no bad local valleys. arXiv preprint arXiv:1809.10749, 2018.

[74] S. Nowozin, B. Cseke, and R. Tomioka. f-gan: Training generative neural samplers using variational divergence minimization. In NeurIPS, 2016.

[75] B. Poole, A. A. Alemi, J. Sohl-Dickstein, and A. Angelova. Improved generator objectives for gans. arXiv preprint arXiv:1612.02780, 2016.

[76] A. Radford, L. Metz, and S. Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. In ICLR, 2016.

[77] S. J. Reddi, S. Kale, and S. Kumar. On the convergence of adam and beyond. In ICLR, 2018.

[78] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, X. Chen, and X. Chen. Improved techniques for training gans. In NeurIPS, 2016.

[79] M. Sanjabi, J. Ba, M. Razaviyayn, and J. D. Lee. On the convergence and robustness of training gans with regularized optimal transport. In NeurIPS, 2018.

[80] R. Sun, D. Li, S. Liang, T. Ding, and R. Srikant. The global landscape of neural networks: An overview. IEEE Signal Processing Magazine, 37(5):95–108, 2020.

[81] R.-Y. Sun. Optimization for deep learning: An overview. Journal of the Operations Research Society of China, pages 1–46, 2020.

[82] D. Tran, R. Ranganath, and D. M. Blei. Deep and hierarchical implicit models. In NeurIPS, 2017.

[83] T. Unterthiner, B. Nessler, C. Seward, G. Klambauer, M. Heusel, H. Ramsauer, and S. Hochreiter. Coulomb gans: Provably optimal nash equilibria via potential fields. In International Conference on Learning Representations, 2018.

[84] L. Venturi, A. S. Bandeira, and J. Bruna. Spurious valleys in two-layer neural network optimization landscapes. arXiv preprint arXiv:1802.06384, 2018.

[85] X. Wang, K. Yu, S. Wu, J. Gu, Y. Liu, C. Dong, Y. Qiao, and C. Change Loy. Esrgan: Enhanced super-resolution generative adversarial networks. In ECCV, 2018.

[86] J. Wu, Z. Huang, W. Li, J. Thoma, and L. Van Gool. Sliced wasserstein generative models. In CVPR, 2019.

[87] Y. Xiangli, Y. Deng, B. Dai, C. C. Loy, and D. Lin. Real or not real, that is the question. arXiv preprint arXiv:2002.05512, 2020.

[88] Y. Yazici, C.-S. Foo, S. Winkler, K.-H. Yap, G. Piliouras, and V. Chandrasekhar. The unusual effectiveness of averaging in gan training. In ICLR, 2019.

[89] H. Zhang, I. Goodfellow, D. Metaxas, and A. Odena. Self-attention generative adversarial networks. In ICMIL, 2018.

[90] J. Zhang, P. Xiao, R. Sun, and Z.-Q. Luo. A single-loop smoothed gradient descent-ascent algorithm for nonconvex-concave min-max problems. arXiv preprint arXiv:2010.15768, 2020.

[91] Y. Zhang, P. Liang, and M. Charikar. A hitting time analysis of stochastic gradient langevin dynamics. arXiv preprint arXiv:1702.05575, 2017.

[92] D. Zou, Y. Cao, D. Zhou, and Q. Gu. Stochastic gradient descent optimizes over-parameterized deep relu networks. arXiv preprint arXiv:1811.08888, 2018.
Appendix: Towards a Better Global Loss Landscape of GANs

The code is available at https://github.com/AilsaF/RS-GAN. This appendix consists of additional experiments, related work, proofs, other results and various discussions.

Contents

1 Introduction

2 Difference of Population Loss and Empirical Loss

3 Landscape Analysis of GANs: Intuition and Toy Results

4 Main Theoretical Results
   4.1 Landscape Results in Function Space
   4.2 Landscape Results in Parameter Space
   4.3 Discussion of Implications

5 Case Study of Two-Cluster Experiments

6 Real Data Experiments

7 Conclusion

A Related Work
   A.1 Related Works on Local Minima and Mode Collapse

B 2-Cluster Experiments: Details and More Discussions

C Result and Experiments for Imbalanced Data Distribution
   C.1 Imbalanced Data: Math Results for Two-Clusters
   C.2 Experiments

D Experiments of Bad Initialization

E Experiments of Regular Training: More Details and More Results
   E.1 Experiment Details and More Experiments with Logistic Loss
   E.2 Experiments with Hinge Loss
   E.3 Experiments with Least Square Loss

F Experiments on High Resolution Data

G Discussions on Empirical Loss and Population Loss (complements Sec. 2)
   G.1 Particle space or probability space?
   G.2 Empirical loss and population loss
   G.3 Generalization and overfitting of GAN

H Proofs for Section 3 (2-Point Case) and Appendix C (2-Cluster Case)
   H.1 Proof of Claim 3.1 and Corollary 3.1 (for JS-GAN)
   H.2 Proof of Claim 3.2 (for RS-GAN)
   H.3 Proofs for 2-Cluster Data (Possibly Imbalanced)

I Proof of Theorem 1 (Landscape of Separable-GAN)

J Proof of Theorem 2 (Landscape of RpGAN)
   J.1 Warm-up Example
   J.2 Proof of Theorem J.1
      J.2.1 Graph Preliminaries and Proof of Lemma J.1
      J.2.2 Proof of Claim J.1
   J.3 Proof of Theorem 2

K Results in Parameter Space
   K.1 Sufficient Conditions for the Assumptions
   K.2 Other Sufficient Conditions
   K.3 Proofs of Propositions for Parameter Space
   K.4 A technical lemma
   K.5 Proof of claims

L Discussion of Wasserstein GAN
A Related Work

We provide a more detailed overview of related work in this section.

Global analysis in supervised learning. Recently, global landscape analysis has attracted much attention. See Sun [81], Sun et al. [80], Bianchini and Gori [15] for surveys and [55, 57, 26, 56, 38, 2, 92, 27] for some recent works. It is widely believed that wide networks have a nice loss landscape and thus local minima are less of a concern (e.g., [60, 32, 50]). However, this claim only holds for supervised learning, and it is not clear whether local minima cause training difficulties for GANs.

Single-mode analysis. For single-mode data, Feizi et al. [30] and Mescheder et al. [65] provide a global analysis of GANs. They consider a single point 0 and a single Gaussian respectively. Feizi et al. [30] differs from ours in a few aspects. First, they consider the single-mode setting which does not have an issue of mode collapse. Second, they assume $p_{data}$ is a Gaussian distribution, while we consider an arbitrary empirical distribution. Third, they analyze “quadratic-GAN,” which is not common in practice, while we analyze commonly used GAN formulations (including JS-GAN).

Mode collapse. Mode collapse is one of the major challenges for GANs which received a lot of attention. There are a few high-level hypotheses, such as improper loss functions [3, 5] and weak discriminators [66, 78, 5, 52]. Interestingly, RpGAN both changes the loss function and improves the discriminator. The theoretical analysis of mode collapse is relatively scarce. Lin et al. [58] makes a key observation that two distributions with the same total variation (TV) distance to true distribution do not exhibit the same degree of mode collapse. They proposed to pack the samples (PacGAN) to alleviate mode collapse. This work is rather different from ours. First, they analyze the TV distance, while we analyzed SepGANs and RpGANs. Second, their analysis is statistical, while our analysis is about optimization. As for the empirical guidance, RpGAN and PacGAN are complimentary and can be used together (suggested by the author of [41]). There are a few more works that discuss mode collapse and/or local minima; we defer the discussion to Appendix A.1.

Theoretical studies of loss functions. The early work on GANs [35] built a link between the min-max formulation and the J-S distance to justify the formulation. Arjovsky and Bottou [3] pointed out some possible drawbacks of J-S distance, and proposed a new loss based on Wasserstein distance, referred to as WGAN. Later, Arora et al. [5] point out that both Wasserstein distance and J-S distance are not generalizable, but they also argued that this is not too scary since people are not directly minimizing these two distances but a class of metrics referred to as “neural-network distance.”

Convergence analysis. Many recent works analyze convergence of GANs and/or min-max optimization, e.g., [23, 22, 6, 34, 64, 88, 39, 79, 90]. These works often only analyze local stability or convergence to local minima (or stationary points), making it different from our work. Lei et al. [48] studied the convergence of WGAN, but restricted to 1-layer neural nets.

Other theoretical analysis. There are a few other theoretical analysis of GANs, e.g., [68, 59, 29, 16, 8, 51, 61, 48]. Most of these works are not directly related to our work.

Other GAN Variants. There are many GAN variants, e.g., WGAN [4] [3, 36] and variants [86, 46, 1, 24, 25], f-GAN [74], SN-GAN [67], self-attention GAN [89], StyleGAN [43, 44] and many more [63, 69, 12, 70, 21, 54, 49, 78, 74, 75, 66, 37, 76, 10, 49]. Our analysis framework (analyzing global landscape of empirical loss) can potentially be applied to more variants mentioned above.

A.1 Related Works on Local Minima and Mode Collapse

We discuss a few related works on local minima and mode collapse, including Kodali et al. [45], Li and Malik [53] and Unterthiner et al. [83] that are mentioned in the main text.

DRAGAN. Kodali et al. [45] suggested the connection between mode collapse and a bad equilibrium based on the following empirical observation: a sudden increase of the gradient norm of the discriminator during training is associated with a sudden drop of the IS score. However, Kodali et al. [45] don’t present formal theoretical results on the relation between mode collapse and a bad equilibrium.

IMLE. Li and Malik [53] proposed implicit maximum likelihood estimation (IMLE). The empirical version of IMLE in the parameter space is the following:

$$\min_{w} \sum_{j=1}^{n} \min_{i \in [1, \ldots, m]} \left\| x_i - G_w(z_j) \right\|^2. \quad (9)$$
In other words, for each generated sample \( y_j = G_w(z_j) \), the loss is the distance from \( y_j \) to the closest true sample \( x_i \). Interestingly, IMLE and RpGAN both couple the true data and the fake data in the loss. The differences are two fold: first, IMLE does not have an extra discriminator \( f_0 \), while RpGAN has; second, IMLE compares \( y_j \) with all \( x_i \) (so as to find the nearest neighbor) while RpGAN compares \( y_j \) with an arbitrary \( x_j \). See Table 3 for a comparison. Note that Li and Malik [53] don’t present formal theoretical results on the landscape.

| Model name | Empirical form of loss | Form of coupling | Optimization |
|------------|------------------------|------------------|--------------|
| RpGAN [41] | \( \max_f \sum_j h(f(x_j) - f(y_j)) \) | pairing | min-max \(^a\) |
| RaGAN [41] | \( \max_f \sum_j h(\frac{1}{n} \sum_{i=1}^n f(x_i) - f(y_j)) \) | comparing with average | min-max |
| (max-)sliced-WGAN [24][25] | \( \max_{|f| \leq 1} \sum_i [f(X)_i - f(Y)_i]^2 \) | pairing sorted output | min-max |
| IMLE [53] | \( \sum_{i,j} k(x_i, x_j) + \sum_{i,j} k(y_i, y_j) \) | all-pairs | non-zero-sum game \(^vi\) |
| Coulomb-GAN [83] | \( \sum_{i,j} k(x_i, x_j) + \sum_{i,j} k(y_i, y_j) - 2 \sum_{i,j} k(x_i, y_j) \) | \(^v\) | |

\(^a\) We show the empirical form of the loss in the function space. Rigorously speaking, the provided form is the the loss for one mini-batch; in practice, in different iterations of SGD we will use different samples of \( x_i, y_j \). For the empirical loss in the parameter space, we shall replace \( f \) by \( f_0 \) and \( y_j \) by \( G_w(z_j) \).

\(^v\) Besides the zero-sum game form (min-max form), RpGAN can be easily modified to a non-zero-sum game form (“non-saturating version” proposed in [25]). The precise expression of RaGAN (relativistic averaging GAN) shall be \( \sum_i h_2(\frac{1}{n} \sum_{i=1}^n f_0(x_i) - f_0(y_j)) + \sum_i h_2(\frac{1}{n} \sum_{j=1}^n f_0(y_j) - f_0(x_i)) \). For simplicity we only present one term in the table.

\(^vi\) Here \( k \) is the Coulomb kernel, defined as \( k(u, v) = \frac{1}{(\sqrt{|u-v|^2 + \epsilon^2})^\alpha} \), where \( u, v \in \mathbb{R}^d, \alpha \leq d - 2 \) and \( \epsilon > 0 \). The original form of Coulomb-GAN is a non-zero-sum game, but it is straightforward to transfer the formulation to a pure minimization form since the discriminator-minimization problem has a closed form solution (used in the proof of [83] Theorem 2). We presented the transformed minimization problem here.

Coulomb-GAN. Unterthiner et al. [83] argued that mode collapse can be a local Nash equilibrium in an example of two clusters (see [83] Appendix A.1). They further proposed CoulombGAN and claimed that every local Nash equilibrium is a global Nash equilibrium (see [83] Theorem 2). Their study is different from ours in a few aspects. First, they still consider the pdf \( p_g \), though restrict the possible movement of \( p_g \) (according to a continuity equation). In contrast, we consider the empirical loss in particle space. Second, the bad landscape of JS-GAN is discussed in words for the 2-cluster case [83] Appendix A.1, but not formally proved. In contrast, we prove rigorous result for the general case. Third, they do not study parameter space (though with informal discussion). Fourth, they do not present landscape-related experiments, such as the narrow-net experiments we have done.

**Common idea: Coupling true data and fake data.** Interestingly, similar to IMLE andRpGAN, CoulombGAN also coupled the true data and fake data in the loss functions. RpGAN, RaGAN (a variant of RpGAN considered in [41]), IMLE and CoulombGAN differ in two aspects: the specific form of coupling (pairing, comparing with average, comparing with the closest, all possible pairs), and the specific form of optimization (pure minimization, min-max, non-zero-sum game). See the comparison in Table 3. It is interesting that all three lines of work choose to couple true data and fake data to resolve the issue of mode collapse. We suspect it is hard to prove similar results on the landscape of empirical loss for IMLE and Coulomb-GAN.

**Relation to (max)-sliced Wasserstein GAN.** We point out that the sliced Wasserstein GAN (sliced-WGAN) [24] and the max-sliced Wasserstein GAN (max-sliced-WGAN) [25] also couple the true data and fake data. For any function \( f \), denote \( f(X) = (f(x_1), \ldots, f(x_n)) \) and \( f(Y) = (f(y_1), \ldots, f(y_n)) \). The empirical version of the max-sliced Wasserstein GAN can be written as

\[
\min_Y \max_{|f| \leq 1} W_2(f(X), f(Y))^2. \tag{10}
\]

Here \( f \) is a neural net with codomain \( \mathbb{R} \), and \( W_2 \) is the Wasserstein-2-distance. Denote \( f(X)_{(i)} \leq \cdots \leq f(X)_{(n)} \) and \( f(Y)_{(1)} \leq \cdots \leq f(Y)_{(n)} \) as the sorted versions of \( f(x_i) \)’s and \( f(y_i) \)’s respectively.
B 2-Cluster Experiments: Details and More Discussions

In this part, we present details of the experiments in Section 5 and other complementary experiments.

**Experimental Setting.** The code is provided in “GAN_2Cluster.py”. We sample 100 points from two clusters of data near 0 and 4 (roughly 50 in each cluster). We use GD with momentum parameter 0.9 for both D and G. The default learning rate is (Dlr, Glr) = (10^{-2}, 10^{-2}). The default inner-iteration-number for the discriminator and the generator are (DIter, GIter) = (10, 10). The discriminator and generator are a 4-layer network (with 2 hidden layers) with sigmoid activation and tanh activation respectively. The default neural network width (Dwidth, Gwidth) = (10, 5). We will also discuss the results of other hyperparameters. The default number of training iterations is MaxIter = 5000. We use the non-saturating versions for both JS-GAN and RS-GAN.

**Understanding the effect of mode collapse, by checking D loss evolution and data movement.** In the main text, we discussed that mode collapse can slow down training of JS-GAN. For easier understanding of the training process, we add the visualization of the data movement (which is possible since we are dealing with 1-dimensional data) in Figure 7. We use the y-axis to denote the data position, and x-axis to denote the iteration. The blue curves represent the movement of all fake data during training, and the red straight lines represent the position of true data (two clusters). The training time may vary across different runs, but overall the time for JS-GAN is about 2-4 times longer than that for RS-GAN.

**Effect of width.** The default width is (Dwidth, Gwidth) = (10, 5). We tested two other settings: (20, 10) and (5, 3). For the wide-network setting, the convergence of both JS-GAN and RS-GAN are much faster, but RS-GAN is still faster than JS-GAN in most cases; see Fig. 8. For the narrow-network setting, RS-GAN can recover two modes in all five runs, while JS-GAN fails in two of the five runs (within 5k iterations). See Fig. 9 for one success case of JS-GAN and one failure case of RS-GAN. In the failure case, JS-GAN completely gets stuck at mode collapse, and the D loss is stuck at around 0.48, consistent with our theory.

---

Note that max-sliced-WGAN in Deshpande et al. [23] uses \( \min_y \max_{|v|\leq 1} W_2(v^T g(X), v^T g(Y))^2 \), while sliced-WGAN in Deshpande et al. [23] uses \( \min_y \max_{|v|\leq 1} W_2(v^T g(X), v^T g(Y))^2 \). In Eq. (10) we use \( f(u) \) to replace \( v^T g(u) \) to simplify the expression; although technically, \( f \) and \( v^T g \) are not equivalent, this minor difference does not affect our discussion.
Other hyperparameters. Besides the width, the learning rates and (DIter, GIter) will also affect the training process. As for (DIter, GIter), we use (10, 10) as default, but other choices such as (5, 2) and (1, 1) also work. As for learning rates, we use (0.01, 0.01) as default, but smaller learning rates such as (0.001, 0.001) also work. Different from the default hyper-parameters, for some hyper-parameters, the D loss of JS-GAN does not reach 0.48, indicating that the basin only attracts the iterates half-way. Nevertheless, in most settings RS-GAN is still faster than JS-GAN.

C Result and Experiments for Imbalanced Data Distribution

In the main results, we assume \(x_i\)’s are distinct. In this section, we allow \(x_i\)’s to be in general positions, i.e., they can overlap. The 2-point model can only approximate two balanced clusters. Allowing \(x_i\)’s to overlap, we are able to analyze imbalanced two clusters. We will show: (i) a theoretical result for 2-cluster data; (ii) experiments on imbalanced 2-cluster data and MNIST.

C.1 Imbalanced Data: Math Results for Two-Clusters

Assume there are \(n\) true data points \(X = (x_1, \ldots, x_n)\) in two modes with proportion \(\alpha\) and \(1 - \alpha\) respectively, where \(\alpha > 0.5\). More precisely, assume \(x_1 = x_2 = \cdots = x_{n_0}\) and \(x_{n_0+1} = \cdots = x_n\), and denote two multi-sets \(X_1 = \{x_1, x_2, \ldots, x_{n_0}\}\) and \(X_2 = \{x_{n_0+1}, x_2, \ldots, x_n\}\). Denote \(Y = \{y_1, \ldots, y_n\}\) as the tuple of all generated points, and let \(\mathcal{Y}\) be the multiset \(\{y_1, \ldots, y_n\}\).

Claim C.1. Consider the JS-GAN loss defined in Eq. (1), where \(X\) is defined above. We have

\[
\phi_{\text{JS}}(Y, X) = q_{\alpha}(m_1) + q_{1-\alpha}(m_2), \quad \text{if } |X_1 \cap \mathcal{Y}| = m_1, \quad |X_2 \cap \mathcal{Y}| = m_2,
\]

where \(q_{\alpha}(m) = \frac{\alpha}{2} \log(\alpha n) + \frac{m}{2n} \log m - \frac{\alpha n + m}{2n} \log(\alpha n + m)\).  

(12)

As a result, the global minimal loss is \(-\log 2\), which is achieved iff \(\mathcal{Y} = X_1 \cup X_2\).

Corollary C.1. Suppose \(\hat{Y} = (\hat{y}_1, \ldots, \hat{y}_n)\) satisfies \(|X_1 \cap \hat{\mathcal{Y}}| = n_1, \quad |X_2 \cap \hat{\mathcal{Y}}| = n_2\), where \(\hat{\mathcal{Y}} = \{\hat{y}_1, \ldots, \hat{y}_n\}\) is the multiset of all \(\hat{y}_i\)’s, then \(\hat{\mathcal{Y}}\) is a strict local minimum. Moreover, if \(n_1 \neq n\alpha\), then \(\hat{\mathcal{Y}}\) is a sub-optimal strict local minimum.

The proofs of Claims C.1 and Corollary C.1 are given in Appendix H.3.

Denote \(m_1 \triangleq |X_2 \cap \mathcal{Y}|, m_2 \triangleq |X_1 \cap \mathcal{Y}|\). The value \(q_{\alpha}(n)\) indicates the value of \(\phi(Y, X)\) at the mode collapsed pattern (state 1a) where \(m_1 = n, m_2 = 0\). Note that \(q_{\alpha}(n) = \frac{2}{\alpha} \log \frac{\alpha}{n+1} + \frac{1}{2} \log \frac{\alpha}{n+1} + \frac{1}{2} \log \frac{\alpha}{n+1} + \frac{1}{2} \log \frac{\alpha}{n+1} \approx -0.4774\); when \(\alpha = 2/3\), \(q_{\alpha}(n) \approx -0.5608\). The more imbalanced the data are (larger \(\alpha\)), the smaller \(q_{\alpha}(n)\), and
We suspect that the deeper basin in the imbalanced case will make it harder to escape mode collapse while JS-GAN easily gets stuck at mode collapse for imbalanced data. The above two experiments verify our earlier prediction that RS-GAN is robust to imbalanced data.

C.2 Experiments

2-Cluster Experiments. For the balanced case, the experiment is described in Appendix B. Both JS-GAN and RS-GAN can converge to the two-mode-distribution. For the imbalanced case where $\alpha = 2/3$, with other hyper-parameters unchanged, JS-GAN falls into mode collapse while RS-GAN generates the true distribution (2/3 in mode 1 and 1/3 in mode 2) (see Fig. 11). The loss $\phi_{JS}(Y, X)$ ends up at approximately -0.56, which matches Claim C.1.

MNIST experiments. To ease visualization, we create an MNIST sub-dataset only containing 5’s and 7’s. We use the CNN structure of Tab. 7 and train for 30k iterations. For the balanced case, the number of 5’s and 7’s are identical (i.e., ratio 1:1). Both JS-GAN and RS-GAN generate a roughly equal number of 5’s and 7’s, as shown in Fig. 12(a,b). For the imbalanced case with 4 times more 7’s than 5’s (ratio 1:5), JS-GAN only generates 7’s, while RS-GAN generates 13 5’s among 64 generated samples, aligning with the true data distribution (see Fig. 12(c,d)).

The above two experiments verify our earlier prediction that RS-GAN is robust to imbalanced data while JS-GAN easily gets stuck at mode collapse for imbalanced data.

D Experiments of Bad Initialization

A bad optimization landscape does not mean the algorithm always converges to bad local minimum. A ‘bad’ landscape means is that there exists a “bad” initial point (the blue point in Fig. 3(a)) that it will lead to a ‘bad’ final solution upon training. In contrast, a good landscape is more robust to the initial point: starting from any initial point (e.g., two points shown in Fig. 3(b)), the algorithm can still find a good solution. Therefore, bad optimization landscape of JS-GAN does not mean the performance of JS-GAN is bad for any initial point, but it should imply that JS-GAN is bad for certain initial points.

Next, we will show experiments that support this prediction.
Figure 12: Balanced and Imbalanced MNIST setting: Comparison of JS-GAN and RS-GAN.

5-Gaussian Experiments. We consider a 2-dimensional 5-Gaussian distribution as illustrated in Fig. 14(a). We design a procedure to find an initial discriminator and generator. For JS-GAN or RS-GAN, in some runs we obtain mode collapse and in some runs we obtain perfect recovery. Firstly, for the runs achieving perfect recovery (Fig. 14(b)) in JS-GAN and RS-GAN respectively, we pick the generators at the converged solution, which we denote as $G_{JS0}$ and $G_{RS0}$ respectively. Secondly, for the runs attaining mode collapse (Fig. 14(c)) in JS-GAN and RS-GAN respectively, we pick the discriminators at the converged solution, referred to as $D_{JS0}$ and $D_{RS0}$. Then we re-train both JS-GAN and RS-GAN from $(D_{JS0}, G_{JS0})$ and $(D_{RS0}, G_{RS0})$ respectively.

We define an evaluation metric $\Psi = \sum_{k=1}^{K} \min_{1 \leq i \leq 10^4} (\alpha \|x_i - C_k\|)$, where $C_k$’s are the cluster centers, $\alpha$ is a scalar and $x_i$’s are $10^4$ true data samples. We repeat the experiment $S = 50$ times and compute the average $\Psi$. The larger the metric, the worse the generated points. As shown in Fig. 14(a), the metric $\Phi$ is much higher for JS-GAN than for RS-GAN, for various learning rates $lr$.

MNIST Experiments. We use a similar strategy to find initial parameters for MNIST data. Fig. 15 (also in Sec. 6) shows that RS-GAN generates much lower FID scores (30+ gap) than JS-GAN. The two experiments verify our prediction that RS-GAN is more robust to initialization, which supports our theory that RS-GAN enjoys a better landscape than JS-GAN.

E Experiments of Regular Training: More Details and More Results

In this section, we present details of the regular experiments in Sec. 6 and a few more experiments.

E.1 Experiment Details and More Experiments with Logistic Loss

Non-saturating version. Following the standard practice [35], if $\lim_{t \to \infty} h(t) = 0$, we use the non-saturating version of RpGAN in practical training:

$$
\min_{\theta} L_D(\theta; w) \triangleq \frac{1}{n} \sum_i h(f_{\theta}(x_i) - f_{\theta}(G_w(z_i))),
$$

$$
\min_{w} L_G(w; \theta) \triangleq \frac{1}{n} \sum_i h(f_{\theta}(G_w(z_i)) - f_{\theta}(x_i)).
$$

(13)

For logistic and hinge loss, we use Eq. (13). For least-square loss, we use the original min-max version (check Appendix E.3 for more). We use alternating stochastic GDA to solve this problem.

Neural-net structures: We conduct experiments on two datasets: CIFAR-10 (32 × 32 size) and STL-10 (48 × 48 size) on both standard CNN and ResNet. As mentioned in Sec. 6, we also conduct experiments on the narrower nets: we reduce the number of channels for all convolutional layers in the generator and discriminator to (1) half, (2) quarter and (3) bottleneck (for ResNet structure),
For SepGAN, we set glr = 0.0002 for SepGANs (JS-GAN, hinge-GAN) as suggested by \( w \). For RpGANs, we find that the learning rate for the generator (glr) needs to be larger (generator lr = discriminator lr).

Generated samples obtained upon training on STL-10 dataset are given in Fig. 16 for CNN, Fig. 17 for ResNet. Generated samples obtained upon training on CIFAR-10 are given in Fig. 18 for CNN, Fig. 19 for ResNet. Instead of cherry-picking, all sample images are generated from random sampled Gaussian noise.

### Table 4: Repeat the experiments (logistic loss) in Tab. 2 with at least three seeds.

| ResNet  | IS ↑ | FID ↓ | IS ↑ | FID ↓ | IS ↑ | FID ↓ |
|---------|------|-------|------|-------|------|-------|
| JS-GAN+SN  | 8.03±0.10 20.06±0.18  | 8.41±0.09 17.79±0.43  | 9.14±0.12 33.06  |
| RS-GAN+SN  | 7.94±0.09 19.79±0.57  | 8.37±0.10 17.75±0.56  | 9.23±0.08 31.87  |
| JS-GAN+SN+GD channel/2 | 7.77±0.08 23.36±0.46  | 8.24±0.08 20.55±0.59  | 8.69±0.08 42.05  |
| RS-GAN+SN+GD channel/2 | 7.76±0.07 21.63±0.51  | 8.21±0.09 18.91±0.45  | 8.70±0.13 39.31  |
| JS-GAN+SN+GD channel/4 | 6.75±0.06 44.39±4.38  | 7.18±0.06 38.75±6.28  | 8.42±0.06 52.38  |
| RS-GAN+SN+GD feature/4 | 7.20±0.07 31.40±0.78  | 7.60±0.06 26.85±0.56  | 8.43±0.10 48.92  |
| JS-GAN+SN+BottleNeck  | 7.51±0.07 27.33±1.05  | 7.99±0.10 23.71±0.86  | 8.37±0.08 47.97  |
| RS-GAN+SN+BottleNeck  | 7.52±0.10 25.05±0.35  | 8.06±0.11 21.29±0.22  | 8.48±0.06 44.60  |

### Hyper-parameters:
- We use a batchsize of 64. For CIFAR-10 on ResNet we set \( \beta_1 = 0 \) and \( \beta_2 = 0.9 \) in Adam. For others, \( \beta_1 = 0.5 \) and \( \beta_2 = 0.999 \). We use GIter = 1 for both CNN and ResNet. We also use DIter = 1 for CNN and DIter = 5 for ResNet. We fix the learning rate for the discriminator (dlr) to be 2e-4. For RpGANs, we find that the learning rate for the generator (glr) needs to be larger than dlr to keep the training balanced. Thus we tune glr using parameters in the set 2e-4, 5e-4, 1e-3, 1.5e-3. For SepGAN, we set glr = 0.0002 for SepGANs (JS-GAN, hinge-GAN) as suggested by \[67, 76\].

### More details of EMA:
- In Sec. 6 we conjectured that the effect of EMA (exponential moving average) and RpGAN are additive. Suppose \( w^{(t)} \) is the generator parameter in \( t \)-th iteration of one run, the EMA generator at the \( t \)-th iteration is computed as follows
  \[
  w_{\text{EMA}}^{(t)} = \beta w_{\text{EMA}}^{(t-1)} + (1-\beta) w^{(t)},
  \]
  where \( w_{\text{EMA}}^{(0)} = w^{(0)} \). Note that EMA is a post-hoc processing step, and does not affect the training process. Intuitively, the EMA generator is closer to the bottom of a basin while the real training is circling around a basin due to the minmax structure. We set \( \beta = 0.9999 \). As Tab. 3 shows, while EMA improves both JS-GAN and RS-GAN, RS-GAN is still better than JS-GAN.

### Results on Logistic Loss with More Seeds:
- Besides the result in Tab. 2 we run at least 3 extra seeds for all experiments with ResNet structure on CIFAR-10 to show that the results are consistent across different runs. We report the results in Tab. 4 and find RS-GAN is still better than JS-GAN and the gap increases as the networks become narrower.

### Samples of image generation:
- Generated samples obtained upon training on CIFAR-10 are given in Fig. 16 for CNN, Fig. 17 for ResNet. Generated samples obtained upon training on STL-10 dataset are given in Fig. 18 for CNN, Fig. 19 for ResNet. Instead of cherry-picking, all sample images are generated from random sampled Gaussian noise.

---

\[7\] We tuned glr in the set 2e-4, 5e-4, 1e-3, 1.5e-3 and find that glr = 2e-4 performs the best in most cases for SepGAN, so we follow the suggestion of \[67, 76\].
We consider the least square loss. The LS-GAN [62] is defined as follows:

\[
\min_{\theta} L_D^{LS}(\theta; w) \triangleq \frac{1}{2n} \left[ \sum_i \max(0, 1 - D_{\theta}(x_i)) + \sum_i \max(0, 1 + D_{w}(G_{\theta}(z_i))) \right],
\]

\[
\min_{w} L_G^{LS}(w; \theta) \triangleq \frac{1}{n} \sum_i \left[ f_{\theta}(G_{\theta}(z_i)) - f_{w}(x_i) \right].
\]

Table 5: Comparison of Hinge-GAN and Rp-Hinge-GAN. We also show the FID gap between Rp-Hinge-GAN with Hinge-GAN (e.g. 2.20 = 21.30 − 19.10 and 9.96 = 33.24 − 23.28).

| ResNet + Hinge Loss | CIFAR-10 | CIFAR-10 + EMA |
|---------------------|----------|----------------|
|                     | IS ↑     | FID ↓ | FID Gap | IS ↑ | FID ↓ | FID Gap |
| Hinge-GAN           | 7.92±0.08 | 21.30 | 8.44±0.10 | 17.43 |
| Hinge-GAN +GD channel/2 | 7.63±0.05 | 27.21 | 7.90±0.08 | 24.35 |
| Hinge-GAN +GD channel/4 | 6.79±0.09 | 37.51 | 7.39±0.07 | 34.45 |
| Hinge-GAN +BottleNeck | 7.16±0.10 | 33.24 | 7.91±0.09 | 26.56 |
| Rp-Hinge-GAN        | 7.84±0.09 | 19.10 | 2.20 | 8.21±0.09 | 17.19 | 0.24 |
| Rp-Hinge-GAN +GD channel/2 | 7.77±0.08 | 21.10 | 6.11 | 8.34±0.11 | 19.19 | 5.17 |
| Rp-Hinge-GAN +GD channel/4 | 7.21±0.11 | 29.41 | 8.10 | 7.77±0.08 | 25.57 | 8.88 |
| Rp-Hinge-GAN +BottleNeck | 7.52±0.07 | 23.28 | 9.96 | 8.05±0.07 | 22.03 | 4.53 |

E.2 Experiments with Hinge Loss

Hinge loss has become popular in GANs [82, 67, 18]. The empirical loss of hinge-GAN is

\[
\min_{\theta} L_D^{Hinge}(\theta; w) \triangleq \frac{1}{2n} \left[ \sum_i \max(0, 1 - D_{\theta}(x_i)) + \sum_i \max(0, 1 + D_{w}(G_{\theta}(z_i))) \right],
\]

\[
\min_{w} L_G^{Hinge}(w; \theta) \triangleq -\frac{1}{n} \sum_i D_{w}(G_{\theta}(z_i)).
\]

Note that Hinge-GAN applies the hinge loss for the discriminator, and linear loss for the generator. This is a variant of SepGAN with \( h_1(t) = -\max(0, 1 - t) \).

The Rp-hinge-GAN is RpGAN given in Eq. (13) with \( h(t) = -\max(0, 1 - t) \):

\[
\min_{\theta} L_D^{R-hinge}(\theta; w) \triangleq \frac{1}{n} \sum_i \max(0, 1 + (f_{\theta}(G_{\theta}(z_i)) - f_{w}(x_i))),
\]

\[
\min_{w} L_G^{R-hinge}(w; \theta) \triangleq -\frac{1}{n} \sum_i \max(0, 1 + (f_{\theta}(x_i) - f_{w}(G_{\theta}(z_i)))�)
\]

We compare them on ResNet with the hyper-parameter settings in Appendix E.1. As Tab. 5 shows, Rp-Hinge-GAN (both versions) performs better than Hinge-GAN. For narrower networks, the gap is 4 to 9 FID scores, larger than the gap for the logistic loss.

E.3 Experiments with Least Square Loss

We consider the least square loss. The LS-GAN [62] is defined as follows:

\[
\min_{\theta} L_D^{LS}(\theta; w) \triangleq \frac{1}{2n} \left[ \sum_i \left( f_{\theta}(x_i) - 1 \right)^2 + \sum_i f_{w}(G_{\theta}(z_i))^2 \right],
\]

\[
\min_{w} L_G^{LS}(w; \theta) \triangleq \frac{1}{n} \sum_i \left( f_{\theta}(G_{\theta}(z_i)) - 1 \right)^2.
\]

This is a non-zero-sum variant of SepGAN with \( h_1(t) = -(1 - t)^2, h_2(t) = -t^2 \).

The Rp-LS-GAN addresses the following objectives:

\[
\min_{\theta} L_D^{R-LS}(\theta; w) \triangleq \frac{1}{n} \sum_i \left( f_{\theta}(x_i) - f_{w}(G_{\theta}(z_i)) - 1 \right)^2,
\]

\[
\min_{w} L_G^{R-LS}(w; \theta) \triangleq -L_D^{R-LS}(\theta; w) = -\frac{1}{n} \sum_i \left( f_{\theta}(x_i) - f_{w}(G_{\theta}(z_i)) - 1 \right)^2.
\]

For least square loss \( h(t) = -(t - 1)^2 \), the gradient vanishing issue due to \( h \) does not exist, thus we can use the min-max version given in Eq. (14) in practice. Our version of Rp-LS-GAN is actually different from the version of Rp-LS-GAN in [41] which is similar to Eq. (13) with least square \( h \).

In Tab. 5 we compare LS-GAN and Rp-LS-GAN on CIFAR-10 with CNN architectures detailed in Tab. 6. As Tab. 5 shows, Rp-LS-GAN is slightly worse than LS-GAN in regular width, but is better than LS-GAN (with 5.7 FID gap) when using 1/4 width.

F Experiments on High Resolution Data

There are two approaches to achieve a good landscape: one uses a wide enough neural net [73, 50], and the other uses a large enough number of samples (approaching convexity of pdf space). As we
discuss in Sec. 2 (see also Appendix G.1), when the number of samples is far from enough for filling
the data space, the convexity (of pdf space) may vanish. A higher dimension of data implies a larger
gap between empirical loss and population loss, thus the non-convexity issue will become more
severe. Thus we conjecture that JS-GAN suffers more for higher resolution data generation.

We consider 256 × 256 LSUN Church and Tower datasets with CNN architecture in Tab. 8. For
RS-GAN, we set glr = 1e-3 and dlr = 2e-4. We train 100,000 iterations with batchsize 64. The
generated images are presented in Fig. 20. For both datasets, RS-GAN outperforms JS-GAN visually.

G Discussions on Empirical Loss and Population Loss (complements Sec. 2)

As mentioned in Sec. 2, the pdf space view (the population loss) was first used in [35], and became
quite popular for GAN analysis. See, e.g., [71, 40, 20]. In this part, we provide more discussions on
the relation of empirical loss and population loss in GANs.

G.1 Particle space or probability space?

Suppose \( p_z = \mathcal{N}(0, I_d_z) \) (or other distributions) is the distribution of the latent variable \( z \), and
\( Z = (z_1, \ldots, z_n) \) are the samples of latent variables. During training, the parameter \( w \) of the
generator net \( G_w \) is moving, and, as a result, both the pdf \( p_g = G_w(p_z) \) and the particles \( y_j = G_w(z_j) \)
move accordingly. Therefore, GAN training can be viewed as either probability space optimization or
particle space optimization. The two views (pdf space and particle space) are illustrated in Figure 1.

In the probability space view, an implicit assumption is that the pdf \( p_g \) moves freely; in the particle
space view, we assume the particles move freely. Free-particle-movement implies free-pdf-movement
if the particles almost occupy the whole space (a one-mode distribution), as shown in Fig. 21.
However, for multi-mode distributions in high-dimensional space, the particles are sparse in the space,
and free-particle-movement does NOT imply free-pdf-movement. This gap was also pointed out in
[83]. Here, we stress that the gap becomes larger for sparser samples (either due to few samples or
high dimension). This forms the foundation for experiments in App. F.

To illustrate the gap between free-pdf-movement and free-particle-movement, we use an example of
learning a two-mode distribution \( p_{\text{data}} \). Suppose we start from an initial two-mode distribution \( p_g \), as
shown Figure 22. To learn \( p_{\text{data}} \), we need to do two things: first, move the two modes of \( p_g \) to roughly
overlap with the two modes of \( p_{\text{data}} \) which we call “macro-learning”; second, adjust the distributions
of each mode to match those of \( p_{\text{data}} \), which we call “micro-learning.” This decomposition is
illustrated in Fig. 22 and 23. In micro-learning, the pdf can move freely, but in macro-learning, the
whole mode has to move together and cannot move freely in the pdf space.

![Figure 21: Illustration of the learning process of the single mode. The generated samples are moving, which corresponds to adjustment of the probability densities.](image1)

![Figure 22: Illustration of the process of learning a multi-mode distribution. We decompose this process into two parts in the next figure.](image2)

![Figure 23: Decomposing learning a multi-mode distribution into macro-learning and micro-learning. Macro-learning refers to the movement of the whole mode towards the underlying data mode. Micro-learning refers to the adjustment of the distribution within each mode. If macro-learning fails, then an entire mode is missed in the generated distributions, which corresponds to mode collapse.](image3)
G.2 Empirical loss and population loss

The population version of RpGAN is
\[
\phi_{R,E}(p_g, p_{\text{data}}) = \sup_{f \in C(R^d)} \mathbb{E}_{(x,y) \sim (p_g, p_{\text{data}})}[h(f(x) - f(y))].
\]  
(15)

Suppose we sample \(x_1, \ldots, x_n \sim p_{\text{data}}\) and \(y_1, \ldots, y_n \sim p_g\), then \(\frac{1}{n} \sum^n_{i=1} [h(f(x_i) - f(y_i))]\) is an approximation of \(\mathbb{E}_{(x,y) \sim (p_g, p_{\text{data}})}[h(f(x) - f(y))]\). The empirical version of RpGAN addresses \(\min_{Y \in \mathbb{R}^{d \times n}} \phi_{R}(Y,X)\), where
\[
\phi_{R}(Y,X) = \sup_{f \in C(R^d)} \frac{1}{n} \sum^n_{i=1} [h(f(x_i) - f(y_i))].
\]  
(16)

Our analysis is about the geometry of \(\phi_{R}(Y,X)\) in Eq. (16). In practical SGDA (stochastic GDA), at each iteration we draw a mini-batch of samples and update the parameters based on the mini-batch. The samples of true data \(x_i\) are re-used multiple times (similar to SGD for a finite-sum optimization), but the samples of latent variables \(z_i\) are fresh (similar to on-line optimization). Due to the re-use of true data, stochastic GDA shall be viewed as an online optimization algorithm for solving Eq. (16) where \(x_i\)’s can be the same. Recall that in the main results, we have assumed that \(x_i\)’s are distinct, thus there is a gap between our results and practice. Extending our results to the case of non-distinct \(x_i\)’s requires extra work. This was done in Claim C.1 for the 2-cluster setting. But for readability we do not further study this setting in the more general cases. We leave this to future work.

G.3 Generalization and overfitting of GAN

One may wonder whether fitting the empirical distribution can cause memorization and failure to generate new data. Arora et al. [5] proved that for many GANs (including JS-GAN) with neural nets, only a polynomial number of samples is needed to achieve a small generalization error. We suspect that a similar generalization bound can be derived for RpGAN.

We provide some intuition why fitting the empirical data distribution via a GAN may avoid overfitting. Consider learning a two-cluster distribution as shown in Fig. 24. During training, we learn a generator that maps the latent samples \(z_i\) to \(x_i\), thus fitting the empirical distribution. If we sample a new latent sample \(z_i\), then the generator will map \(z_i\) to a new point \(x_i\), in the underlying data distribution (due to the continuity of the generator function). Thus the continuity of the generator (or the restricted power of the generator) provides regularization for achieving generalization.

H Proofs for Section 3 (2-Point Case) and Appendix C (2-Cluster Case)

We now provide the proofs for the toy results (i.e., the case \(n = 2\)).

H.1 Proof of Claim 3.1 and Corollary 3.1 (for JS-GAN)

Proof of Claim 3.1 We will compute values of \(\phi_{JS}(Y,X)\) for all \(Y\). Recall \(D\) can be any continuous function with range \((0,1)\). Recall that \(\phi_{JS}(Y,X) = \sup_D \frac{1}{2n} \left[ \sum^n_{i,j=1} \log(D(x_i)) + \sum^n_{i=1} \log(1 - D(y_i)) \right]\). Consider four cases. Denote a multiset \(Y = \{y_1, y_2\}\), and let \(m_i = |Y \cap \{x_i\}|, i \in \{1, 2\}\).

Case 1 (state 1): \(m_1 = m_2 = 1\). Then the objective is
\[
\sup_D \left[ \frac{1}{2} \log(D(x_1)) + \frac{1}{2} \log(1 - D(x_1)) + \frac{1}{2} \log(D(x_2)) + \frac{1}{2} \log(1 - D(x_2)) \right].
\]

The optimal value is \(-\log 2\), which is achieved when \(D(x_1) = D(x_2) = \frac{1}{2}\).

Case 2 (state 1a): \(\{m_1, m_2\} = \{0, 1\}\). WLOG, assume \(m_1 = 1, m_2 = 0\), and \(y_1 = x_1, y_2 \notin \{x_1, x_2\}\). The objective becomes
\[
\sup_D \left[ \frac{1}{2} \log(D(x_1)) + \frac{1}{2} \log(D(x_2)) + \frac{1}{2} \log(1 - D(x_1)) + \frac{1}{2} \log(1 - D(y_2)) \right].
\]
The optimal value \(-\log 2/2\) is achieved when \(D(x_1) = 1/2, D(x_2) \to 1\) and \(D(y_2) \to 0\).

**Case 3** (state 1b): \(\{m_1, m_2\} = \{0, 2\}\). WLOG, assume \(y_1 = y_2 = x_1\). The objective becomes

\[
\sup_D \frac{1}{2} \left[ \log(D(x_1)) + \log(1 - D(x_1)) + \frac{1}{2} \log(D(x_2)) \right].
\]

The optimal value \(\frac{1}{4} \log \frac{1}{3} + \frac{1}{4} \log \frac{3}{2} \approx -0.4774\) is achieved when \(D(x_1) = 1/3\) and \(D(x_2) \to 1\).

**Case 4** (state 2): \(m_1 = m_2 = 0\), i.e., \(y_1, y_2 \notin \{x_1, x_2\}\). The objective is:

\[
\sup_D \frac{1}{2} \left[ \log(D(x_1)) + \frac{1}{2} \log(D(x_2)) + \frac{1}{2} \log(1 - D(x_1)) + \frac{1}{2} \log(1 - D(x_2)) \right].
\]

These terms are independent, thus each term can achieve its supreme \(\log 1 = 0\). Then the optimal value 0 is achieved when \(D(x_1) = D(x_2) \to 1\) and \(D(y_1) = D(y_2) \to 0\).

**Proof of Corollary 3.1** Suppose \(\epsilon\) is the minimal non-zero distance between two points \(x_1, x_2, y_1, y_2\). Consider a small perturbation of \(Y\) as \(Y = (y_1 + \epsilon_1, y_2 + \epsilon_2)\), where \(|\epsilon_i| < \epsilon\). We want to verify that

\[
\phi(\bar{Y}, X) > \phi(Y, X) \approx -0.48.
\]

(17)

There are two possibilities. **Possibility 1**: \(\epsilon_1 = 0\) or \(\epsilon_2 = 0\). WLOG, assume \(\epsilon_1 = 0\), then we must have \(\epsilon_2 > 0\). Then we still have \(y_1 = y_\bar{1} = x_1\). Since the perturbation amount is small enough, we have \(y_2 \notin \{x_1, x_2\}\). According to Case 2 above, we have \(\phi(Y, X) = -\log 2 \approx -0.35 > -0.48\).

**Possibility 2**: \(\epsilon_1 > 0\), \(\epsilon_2 > 0\). Since the perturbation amount \(\epsilon_1\) and \(\epsilon_2\) are small enough, we have \(y_1 \notin \{x_1, x_2\}, y_2 \notin \{x_1, x_2\}\). According to Case 4 above, we have \(\phi(Y, X) > 0 > -0.48\).

Combining both cases, we have proved Eq. (17).

**H.2 Proof of Claim 3.2 for RS-GAN**

This is the result of RS-GAN for \(n = 2\). WLOG, assume \(x_1 = 0, x_2 = 1\). Denote \(g_{\text{RS}}(Y) \triangleq \phi_{\text{RS}}(Y, X) = \sup_{f \in C(R^d)} \frac{1}{2} \log \frac{1}{1 + \exp(f(0) - f(y_1))} + \frac{1}{2} \log \frac{1}{1 + \exp(f(1) - f(y_2))}\). Denote \(m_i = |\{y_i\} \cap \{x_i\}|, i = 1, 2\); note this definition is different from JS-GAN in App. H.4.

**Case 1**: \(m_1 = m_2 = 1\). If \(y_1 = 0, y_2 = 1\), then \(g_{\text{RS}}(Y) = \frac{1}{2} \left[ \log 0.5 + \log 0.5 \right] = -\log 2 \approx -0.6937\). If \(y_1 = 1, y_2 = 0\), then

\[
g_{\text{RS}}(Y) = \sup_{f \in F} \frac{1}{2} \log \frac{1}{1 + \exp(f(0))} + \frac{1}{2} \log \frac{1}{1 + \exp(f(1))} = -\frac{1}{2} \log 2.
\]

**Case 2**: \(\{m_1, m_2\} = \{0, 1\}\). WLOG, assume \(y_1 = 0, y_2 \neq 1\) (note that \(y_2\) can be 0). Then

\[
g_{\text{RS}}(Y) \geq \sup_{f \in F} \frac{1}{2} \log \frac{1}{1 + \exp(f(0) - f(y_2))} + \frac{1}{2} \log \frac{1}{1 + \exp(f(1))} = -\frac{1}{2} \log 2 \approx -0.3466.
\]

The value is achieved when \(f(1) - f(y_2) \to -\infty\).

**Case 3**: \(m_1 = m_2 = 0\). Then

\[
g_{\text{RS}}(Y) \geq \sup_{f \in F} \frac{1}{2} \log \frac{1}{1 + \exp(f(0) - f(y_2))} + \frac{1}{2} \log \frac{1}{1 + \exp(f(1))} = 0.
\]

The value is achieved when \(f(1) - f(y_2) \to -\infty\) and \(f(0) - f(y_2) \to -\infty\).

The global minimal value is \(-\log 2\), and the only global minima are \(\{y_1, y_2\} = \{x_1, x_2\}\). In addition, from any \(Y\), it is easy to verify that there is a non-decreasing path from \(Y\) to a global minimum.
H.3 Proofs for 2-Cluster Data (Possibly Imbalanced)

Proof of Claim [C.1] The proof is built on the proof of Claim [X.1] in Appendix [H.1].

We first consider a special case \(|X_1 \cap \mathcal{Y}|=m_1, |X_2 \cap \mathcal{Y}|=m_2\). This means that \(m\) generated points are in mode 1, and the rest are in neither modes. The loss value can be computed as follows:

\[
\phi_{IS}(Y, X) = \frac{1}{2n} \left[ \alpha n \log(\frac{\alpha n}{\alpha n + m}) + m \log(1 - \frac{\alpha n}{\alpha n + m}) \right] = \frac{\alpha}{2} \log(\alpha n) + \frac{m}{2n} \log m - \frac{\alpha n + m}{2n} \log(\alpha n + m) = q_\alpha(m).
\]

In general, if \(|X_1 \cap \mathcal{Y}|=m_1, |X_2 \cap \mathcal{Y}|=m_2\), then \(\phi_{IS}(Y, X)\) can be divided into three parts: the first part is the sum of the terms that contain \(x_1\) (including \(x_i\)’s and \(y_j\)’s that are equal to \(x_1\)), the second part is the sum of the terms that contain \(x_n\) (including \(x_i\)’s and \(y_j\)’s that are equal to \(x_n\)), and the third part is the sum of the terms that contain \(y_j\)’s that are not in \(\{x_1, x_n\}\). Similar to Case 3 above, the value of the first part is \(q_\alpha(m_1)\), and the value of the second part is \(q_{1-\alpha}(m_2)\). Similar to the above special case, the value of the third part is 0. Therefore, the loss value is \(\phi_{IS}(Y, X) = q_\alpha(m_1) + q_{1-\alpha}(m_2)\).

It is to show that \(q_\alpha(m_1) + q_{1-\alpha}(m_2) \geq - \log 2\), and the equality is achieved iff \(m_1 = n\alpha, m_2 = n(1-\alpha)\), i.e., \(\mathcal{Y} = X_1 \cup X_2\).

Proof sketch of Corollary [C.1] After a small enough perturbation, we must have \(m_1 \triangleq |X_2 \cap \mathcal{Y}| \leq n_1, m_2 \triangleq |X_1 \cap \mathcal{Y}| \leq n_2\). Since \(q_\alpha(m)\) and \(q_{1-\alpha}(m)\) are strictly decreasing functions of \(m\), we have

\[
\phi(Y, X) = q_\alpha(m_1) + q_{1-\alpha}(m_2) \leq q_\alpha(n_1) + q_{1-\alpha}(n_2) = \phi(\hat{Y}, X).
\]

The equality holds iff \((m_1, m_2) = (n_1, n_2)\), i.e., \(Y = \hat{Y}\). This means that if \((n_1, n_2) \neq (n\alpha, n(1-\alpha))\), then \(\hat{Y}\) is a sub-optimal strict local minimum.

We skip the detailed proof, since other parts are similar to the proof of Corollary [3.1]

I Proof of Theorem [T] (Landscape of Separable-GAN)

Denote \(F(D; Y) = \frac{1}{2n} \sum_{i=1}^{n} [h_1(f(x_i)) + h_2(-f(y_i))] \leq 0\) (since \(h_i(t) \leq 0, i = 1, 2\) for any \(t\)).

Step 1: Compute the value of \(\phi(\cdot, X)\) for each \(Y\). For any \(i\), denote \(M_i = \{j : y_j = x_i\}\), \(m_i = |M_i| \geq 0, i = 1, 2, \ldots, n\). Then \(m_1 + \cdots + m_n = n\). Denote \(\Omega = M_1 \cup M_2 \cdots \cup M_n\). Then

\[
\phi(Y, X) = \frac{1}{2n} \sup_{f} \left( \sum_{i=1}^{n} h_1(f(x_i)) + h_2(-f(y_i)) \right) = \frac{1}{2n} \sup_{f} \left( \sum_{i=1}^{n} [h_1(f_i(x_i)) + h_2(-f_i(x_i))] + \|\|f\|\|_2^2 \right)
\]

\[
\leq \frac{1}{2n} \sum_{i=1}^{n} \sup_{t} [h_1(t_i) + m_i h_2(-t_i)] + |Y| \sup_{i \in \Omega} h_2(t) \overset{(i)}{\leq} \frac{1}{2n} \sum_{i=1}^{n} \xi(m_i)
\]

\[
\overset{(ii)}{\leq} \frac{1}{2n} \sum_{i=1}^{n} m_i \xi(1) = \frac{1}{2} \xi(1).
\]

Here (i) is because \(f(y_j), j \in \Omega\) are independent of \(h(x_i)\)’s and thus can be any values; (ii) is by the definition \(\xi(m) = \sup_{t} [h_1(t) + m h_2(-t)]\) and Assumption [4.2] that \(\sup_{t} h_2(t) = 0\); (iii) is due to the convexity of \(\xi\) (note that \(\xi\) is the supreme of linear functions). Furthermore, if there is a certain \(m_i > 1\), then \(\xi(m_i) = (m_i - 1)\xi(0) = \xi(m) > m_i \xi(1)\) (according to Assumption [4.2]), causing (ii) to become a strict inequality. Thus the equality in (ii) holds iff \(m_i = 1, \forall i\), i.e., \(\{y_1, \ldots, y_n\} = \{x_1, \ldots, x_n\}\). Therefore, we have proved that \(\phi(Y, X)\) achieves the minimal value \(\frac{1}{2} \xi(1)\) if \(\{y_1, \ldots, y_n\} = \{x_1, \ldots, x_n\}\).

Step 2: Sufficient condition for strict local-min. Next, we show that if \(Y\) satisfies \(m_1 + m_2 + \cdots + m_n = n\) then \(Y\) is a strict local-min. Denote \(\delta = \min_{k \neq i} \|x_k - x_i\|\). Consider a small perturbation of \(Y\) as \(\tilde{Y} = (\tilde{y}_1, \tilde{y}_2, \ldots, \tilde{y}_n) = (y_1 + \epsilon_1, y_2 + \epsilon_2, \ldots, y_n + \epsilon_n)\), where \(\|\epsilon_j\| < \delta, \forall j\) and \(\sum_j \|\epsilon_j\|^2 > 0\). We want to prove \(\phi(Y, X) > \phi(\tilde{Y}, X)\).

Denote \(\tilde{m}_i = |\{j : \tilde{y}_j = x_i\}|, i = 1, 2, \ldots, n\). Consider an arbitrary \(j\). Since \(y_j \in \{x_1, \ldots, x_n\}\), there must be some \(i\) such that \(\tilde{y}_j = x_i\). Together with \(\|\tilde{y}_j - y_j\| = \|\epsilon_j\| < \delta = \min_{k \neq i} \|x_k - x_i\|\), we have \(y_j \notin \{x_1, x_2, \ldots, x_n\} \backslash \{x_i\}\). In other words, the only possible point in \(\{x_1, \ldots, x_n\}\) that can coincide with \(\tilde{y}_j\) is \(x_i\), and this happens only when \(\epsilon_j = 0\). This implies \(m_i \leq \tilde{m}_i, \forall i\). Since
we have assumed $\sum_j \|e_j\|^2 > 0$, for at least one $i$ we have $\bar{m}_i < m_i$. Together with Assumption 4.3 that $\xi(m)$ is a strictly decreasing function in $m \in [0, n]$, we have $\phi(Y, X) = \frac{1}{n} \sum_{i=1}^n \xi(\bar{m}_i) > \frac{1}{n} \xi(0) = \phi(Y, X)$.

**Step 3: Sub-optimal strict local-min.** Finally, if $Y$ satisfies that $m_1 + m_2 + \cdots + m_n = n$ and $m_k \geq 2$ for some $k$, then $\phi(Y, X) > \frac{1}{n} \xi(0)$. Thus $Y$ is a sub-optimal strict local minimum. Q.E.D.

**Remark 1:** $\xi(m)$ is convex (it is the supreme of linear functions), thus we always have $\xi(m) = \xi(m) + (m - 1)\xi(0) \geq m\xi(1)$. Assump. 4.2 states that the inequality is strict, thus it is slightly stronger than the convexity of $\xi$. By Assump. 4.1, we also have $h_1(t) + (m + 1)h_2(-t) \leq h_1(t) + mh_2(-t)$, thus $\xi(n) \leq \xi(n - 1) \leq \cdots \leq \xi(0)$. Assumption 4.3 states that the inequalities are strict. This holds if the maximizer of $h_1(t) + mh_2(-t)$ does not coincide with the maximizer of $h_2(t)$. Intuitively, if $h(t)$ is “substantially different” from a constant function, then Assump. 4.2 and Assump. 4.3 hold.

**Remark 2:** The upper bound 0 in Assump. 4.1 is not essential, and can be relaxed to any finite numbers (change other two assumptions accordingly). We skip the details.

**J Proof of Theorem 2 (Landscape of RpGAN)**

This proof is the longest one in this paper. We will focus on a proof for the special case of RS-GAN. The proof for general RpGAN is quite similar, and presented in Appendix J.3. Recall $\phi_{\text{RLS}}(Y, X) = \sup_Y f = \frac{1}{n} \sum_{i=1}^n \log (1 + \exp(f(y_i) - f(x_i)))$.

**Theorem J.1. (special case of Theorem 2 for RS-GAN)** Suppose $x_1, x_2, \ldots, x_n \in \mathbb{R}^d$ are distinct. The global minimal value of $\phi_{\text{RS}}(Y, X)$ is $-\log 2$, which is achieved iff $\{x_1, \ldots, x_n\} = \{y_1, \ldots, y_n\}$. Furthermore, any point is global-min-achievable for the function.

**Proof sketch.** We compute the value of $g(Y) = \phi_{\text{RLS}}(Y, X)$ for any $Y$, using the following steps:

(i) We build a graph with vertices representing distinct values of $x_i, y_i$ and draw directed edges from $x_i$ to $y_i$. This graph can be decomposed into cycles and trees.

(ii) Each vertex in a cycle contributes $-\frac{1}{n} \log 2$ to the value $g(Y)$.

(iii) Each vertex in a tree contributes 0 to the value $g(Y)$.

(iv) The value $g(Y)$ equals $-\frac{1}{n} \log 2$ times the number of vertices in the cycles.

The outline of this section is as follows. In the first subsection, we analyze an example as warm-up. Next, we prove Theorem J.1. The proofs of some technical lemmas will be provided in the following subsections. Finally, in Appendix J.3 we present the proof for Theorem 2.

**J.1 Warm-up Example**

We prove that if $\{y_1, y_2, \ldots, y_n\} = \{x_1, \ldots, x_n\}$, then $Y$ is a global minimum of $g(Y)$.

Suppose $y_i = x_{\sigma(i)}$, where $(\sigma(1), \sigma(2), \ldots, \sigma(n))$ is a permutation of $(1, 2, \ldots, n)$. We can divide $\{1, 2, \ldots, n\}$ into finitely many cycles $C_1, C_2, \ldots, C_K$, where each cycle $C_k = (c_k(1), c_k(2), \ldots, c_k(m_k))$ satisfies $c_k(j + 1) = \sigma(c_k(j)), j \in \{1, 2, \ldots, m_k\}$.

Here $c_k(m_k + 1)$ is defined as $c_k(1)$. Now we calculate the value of $g(Y)$:

$$g(Y) = \sup_f \frac{1}{n} \sum_{i=1}^n \log \frac{1}{1 + \exp(f(y_i) - f(x_i)))} \equiv -\inf_f \frac{1}{n} \sum_{k=1}^K \sum_{j=1}^{m_k} \log (1 + \exp(f(y_i) - f(x_i)))$$

$$= -\inf_f \frac{1}{n} \sum_{k=1}^K \sum_{j=1}^{m_k} \log (1 + e^{(s_{c_k(j+1)} - f(s_{c_k(j)}))}) \equiv -\frac{1}{n} \sum_{k=1}^K \sum_{j=1}^{m_k} \inf_t \log (1 + e^{(t_{c_k(j+1)} - f(t_{c_k(j)}))})$$

$$= -\frac{1}{n} \sum_{k=1}^K \sum_{j=1}^{m_k} \inf_t \log (1 + e^{(t_{c_k(j+1)} - t_{c_k(j)}))}) = -\frac{1}{n} \sum_{k=1}^K \sum_{j=1}^{m_k} \log (1 + e^{(t_{c_k(j+1)} - t_{c_k(j)}))})$$

$$= -\frac{1}{n} \sum_{k=1}^K \sum_{j=1}^{m_k} \log (1 + e^{(t_{c_k(j+1)} - t_{c_k(j)}))}) = -\log 2.$$
the choice of \( \{ f(x_i) : i \in C_k \} \) is independent of the choice of \( \{ f(x_i) : i \in C_l \} \) for any \( k \neq l \), thus we can take the infimum over each cycle (i.e., put “inf” inside the sum over \( k \)). (iii) is because \( \sum_{j=1}^{m-1} \log(1 + \exp(t_{j+1} - t_j)) + \log(1 + \exp(t_1 - t_m)) \) is a convex function of \( t_1, t_2, \ldots, t_m \) and the minimum is achieved at \( t_1 = t_2 = \cdots = t_m = 0 \).

### J.2 Proof of Theorem \[ \text{J.1} \]

This proof is divided into three steps. In Step 1, we compute the value of \( g(Y) \) if all \( y_i \in \{ x_1, \ldots, x_n \} \). This is the major step of the whole proof. In Step 2, we compute the value of \( g(Y) \) for any \( Y \). In Step 3, we show that there is a non-decreasing continuous path from \( Y \) to a global minimum.

**Step 1:** Compute \( g(Y) \) that all \( y_i \in \{ x_1, \ldots, x_n \} \). Define

\[
R(X) = \{ Y : y_i \in \{ x_1, \ldots, x_n \}, \forall i \}. \tag{19}
\]

**Step 1.1: Build a graph and decompose it.** We fix \( Y \in R(X) \). We build a directed graph \( G = (V, A) \) as follows. The set of vertices \( V = \{ 1, 2, \ldots, n \} \) represent \( x_1, x_2, \ldots, x_n \). A directed edge \((i, j) \in A \) if \( y_i = x_j \). In this case, there is a term \( \log(1 + \exp(f(x_j) - f(x_i))) \) in \( g(Y) \). It is possible to have a self-loop \((i, i) \), which corresponds to the case \( y_i = x_i \). By Eq. (19), we have

\[
g(Y) = -\inf_f \frac{1}{n} \sum_{i=1}^{n} \log \left( 1 + e^{f(y_i) - f(x_i)} \right) = -\inf_f \frac{1}{n} \sum_{(i,j) \in A} \log \left( 1 + e^{f(x_j) - f(x_i)} \right). \tag{20}
\]

Each \( y_i \) corresponds to a unique \( x_j \), thus the out-degree of \( i \), denoted as \( \text{outdegree}(i) \), must be exactly 1. The in-degree of each \( i \), denoted as \( \text{indegree}(i) \), can be any number in \( \{ 0, 1, \ldots, n \} \).

We will show that the graph \( G \) can be decomposed into the union of cycles and trees (see App. \[ J.2.1 \] for its proof, and definitions of cycles and trees). A graphical illustration is given in Figure 25.

**Lemma 1.** Suppose \( G = (V, A) \) is a directed graph and \( \text{outdegree}(v) = 1, \forall v \in V \). Then:

(a) There exist cycles \( C_1, C_2, \ldots, C_K \) and subtrees \( T_1, T_2, \ldots, T_M \) such that each edge \( v \in A \) appears either in exactly one of the cycles or in exactly one of the subtrees.

(b) The root of each subtree \( u_m \) is a vertex of a certain cycle \( C_k \). In addition, each vertex of the graph appears in exactly one of the following sets: \( V(C_1), \ldots, V(C_K), V(T_1) \setminus \{ u_1 \}, \ldots, V(T_M) \setminus \{ u_M \} \).

(c) There is at least one cycle in the graph.

Figure 25: The first two figures are two connected component of a graph representing the case \( y_i \in \{ x_1, \ldots, x_n \} \), \( \forall i \). The first figure contains 10 vertices and 10 directed edges. It can be decomposed into a cycle (1, 2, 3, 4) and two subtrees: one subtree consists of edges (10, 4) and vertices 10, 4, and another consists of edges (8, 7), (9, 7), (7, 5), (6, 5), (5, 1). The second figure has one cycle being a self-loop, and two trees attached to it. The third figure is an example graph of the case that some \( y_i \notin \{ x_1, \ldots, x_n \} \). In this example, \( n = 8 \) (so 8 edges), and all \( y_i \) are in \( \{ x_1, \ldots, x_n \} \) except \( y_6, y_7 \). The two edges (6, 9) and (6, 9) indicate the two terms \( h(f(y_6) - f(x_6)) \) and \( b(f(y_9) - f(x_9)) \) in \( g(Y) \). They have the same head 9, thus \( y_6 = y_7 \). The vertex 9 has out-degree 0, indicating that \( y_6 = y_7 \notin \{ x_1, \ldots, x_n \} \). This figure can be decomposed into two cycles and three subtrees. Finally, adding a self-loop (9, 9) will generate a graph where each edge has outdegree 1 (this is the reduction done in Step 2).

Denote \( \xi(y_i, x_i) = \log(1 + e^{f(y_i) - f(x_i)}) \). According to Lemma 1, we have

\[
-\inf_f g(Y) = \inf_f \frac{1}{n} \sum_{i=1}^{n} \xi(y_i, x_i) \geq \inf_f \sum_{k=1}^{K} \sum_{i \in V(C_k)} \xi(y_i, x_i) \triangleq g_{\text{cyc}}. \tag{21}
\]

**Step 1.2: Compute \( g_{\text{cyc}} \).** We then compute \( g_{\text{cyc}} \). Since \( C_k \) is a cycle, we have \( X_k \triangleq \{ x_i : i \in C_k \} = \{ y_i : i \in C_k \} \). Since \( C_k \)'s are disjoint, we have \( X_k \cap X_l = \emptyset, \forall k \neq l \). This implies that
Pick a continuous function \( f \). Let \( g \) be a large positive number such that there exists a continuous function \( f \) satisfying

\[
g \geq \inf_{k=1}^{K} \sum_{i \in V(C_k)} \log (1 + \exp(f(y_i) - f(x_i))) = -\log 2 \sum_{k=1}^{K} |V(C_k)|.
\]  (22)

This proves Eq. (24). Combining the two relations given in Eq. (24) and Eq. (23), we have

\[
g(Y) = \inf_{f} F(Y; f) = \frac{1}{n} \sum_{k=1}^{K} |V(C_k)| \log 2, \quad \forall Y \in R(X).
\]  (29)
We present a few definitions from standard graph theory.

\( G = \{ i : y_i \in \{ x_1, \ldots, x_n \} \} \), \( H^c = \{ j : y_j \notin \{ x_1, \ldots, x_n \} \} \).

Since \( y_j \)'s in \( H^c \) may be the same, we define the set of such distinct values of \( y_j \)'s as

\[ Y_{out} = \{ y \in \mathbb{R}^d : y = y_j, \text{ for some } j \in H^c \}. \]

Let \( n = |Y_{out}| \), then there are total \( n + \bar{n} \) distinct values in \( x_1, \ldots, x_n, y_1, \ldots, y_n \). WLOG, assume \( y_1, \ldots, y_{\bar{n}} \) are distinct (this is because the value of \( g(Y) \) does not change if we re-index \( x_i \)'s and \( y_i \)'s as long as the subscripts of \( x_i, y_i \) change together), then

\[ Y_{out} = \{ y_1, \ldots, y_{\bar{n}} \}. \]

We create artificial “true data” and “fake data” \( x_{n+1} = x_{n+1} = y_1, \ldots, x_{n+\bar{n}} = y_{n+\bar{n}} = y_{\bar{n}} \). Define

\[ F_{auc}(Y, f) = -\sum_{i=1}^{n+m} \log \left( 1 + e^{f(x_i) - f(x_{\bar{i}})} \right) \]

\[ g_{auc} = -\inf_f F_{auc}(Y, f). \]

Clearly, \( F_{auc}(Y, f) = n F(Y, f) - \bar{n} \log 2 \) and \( n g(Y) = g_{auc} - \bar{n} \log 2 \).

Consider the new configurations \( \hat{X} = (x_{1}, \ldots, x_{n+\bar{n}}) \) and \( \hat{Y} = (y_1, \ldots, y_{n+\bar{n}}) \). For the new configurations, we can build a graph \( \hat{G} \) with \( n + \bar{n} \) vertices and \( n + \bar{n} \) edges. There are \( K \) self-loops \( C_1, \ldots, C_{K+\bar{n}} \) at the vertices corresponding to \( y_1, \ldots, y_{\bar{n}} \). Based on Lemma \[ \square \] we have: (a) There exist cycles \( C_1, C_2, \ldots, C_{K}, C_{K+1}, \ldots, C_{K+\bar{n}} \) and subtrees \( T_1, T_2, \ldots, T_M \) (with roots \( u_m \)'s) s.t. each edge \( v \in A \) appears in exactly one of the cycle or subtrees. (b) \( u_m \) is a vertex of a certain cycle \( C_k \) where \( 1 \leq k \leq K + \bar{n} \). (c) Each vertex of the graph appears in exactly one of the following sets: \( V(C_1), \ldots, V(C_{K+\bar{n}}), V(T_1) \setminus \{ u_1 \}, \ldots, V(T_M) \setminus \{ u_M \} \). According to the proof in Step 1, we have

\[ n g(Y) = g_{auc} - \bar{n} \log 2 = \sum_{k=1}^{K} |V(C_k)| \log 2 + \bar{n} \log 2. \]

Therefore,

\[ n g(Y) = g_{auc} - \bar{n} \log 2 = \sum_{k=1}^{K} |V(C_k)| \log 2. \]

We build a graph \( G \) by removing the self-loops \( C_{K+\bar{j}} = (y_j, y_j), j = 1, \ldots, \bar{n} \) in \( \hat{G} \). The new graph \( G \) consists of \( n + \bar{n} \) vertices corresponding to \( x_1, \ldots, x_n \) and \( y_1, \ldots, y_{\bar{n}} \) and \( n \) edges. The graph can be decomposed into cycles \( C_1, C_2, \ldots, C_K \) (since \( \bar{n} \) cycles are removed from \( G \)) and subtrees \( T_1, T_2, \ldots, T_M \). The value \( n g(Y) = \sum_{k=1}^{K} |V(C_k)| \log 2 \), where \( \hat{C}_k \)'s are all the cycles of \( G \).

Step 3: Finding a non-decreasing path to a global minimum. Finally, we prove that for any \( Y \), there is a non-decreasing continuous path from \( Y \) to one global minimal \( Y^* \). The following claim shows that we can increase the value of \( Y \) incrementally. See the proof in Appendix J.2.2.

Claim J.1. For an arbitrary \( Y \) that is not a global minimum, there exists another \( \hat{Y} \) and a non-decreasing continuous path from \( Y \) to \( \hat{Y} \) such that \( g(\hat{Y}) - g(Y) \geq \frac{1}{n} \log 2 \).

For any \( Y \) that is not a global minimum, we apply Claim J.1 for finitely many times (no more than \( n \) times), then we will arrive at one global minimum \( Y^* \). We connect all non-decreasing continuous paths and get a non-decreasing continuous path from \( Y \) to \( Y^* \). This finishes the proof.

J.2.1 Graph Preliminaries and Proof of Lemma J.1

We present a few definitions from standard graph theory.

**Definition J.1.** (walk, path, cycle) In a directed graph \( G = (V, A) \), a walk \( W = (v_0, e_1, v_1, e_2, \ldots, v_{m-1}, e_m, v_m) \) is a sequence of vertices and edges such that \( v_i \in V, \forall i \in \{0, 1, \ldots, m\} \) and \( e_i = (v_{i-1}, v_i) \in A, \forall i \in \{1, \ldots, m\} \). If \( v_0, v_1, \ldots, v_m \) are distinct, we call it a path (with length \( m \)). If \( v_0, v_1, \ldots, v_{m-1} \) are distinct and \( v_m = v_0 \), we call it a cycle.

Any \( v \) has a path to itself (with length 0), no matter whether there is an edge between \( v \) to itself or not. This is because the degenerate walk \( W = (v) \) satisfies the above definition. The set of vertices and edges in \( W \) are denoted as \( V(W) \) and \( A(W) \) respectively.
We slightly extend the definition of “walk” to allow infinite length. We present two observations.

**Observation 1:** Starting from any vertex $v_0 \in V(G)$, there is a unique walk with infinite length

$$W(v_0) = (v_0, e_1, v_1, e_2, v_2, \ldots, v_i, e_i, v_i+1, e_{i+1}, \ldots),$$

where $e_i$ is an edge in $A(G)$ with tail $v_{i-1}$ and head $v_i$.

Proof of Observation 1: At each vertex $v_i$, there is a unique outgoing edge $e_i = (v_i, v_{i+1})$ which uniquely defines the next vertex $v_{i+1}$. Continue the process, we have proved Observation 1.

**Observation 2:** The walk $W(v_0)$ can be decomposed into two parts $W_1(v_0) = (v_0, e_1, v_1, e_2, v_2, \ldots, v_i, e_i, v_i+1, e_{i+1}, \ldots)$ and $W_2(v_0) = (v_{i0}, e_{i0+1}, v_{i0+1}, e_{i0+2}, v_{i0+2}, \ldots)$, where $W_1(v_0)$ is a path from $v_0$ to $v_{i0}$ (i.e., $v_0, v_1, \ldots, v_{i0}$ are distinct), and $W_2(v_0)$ is the repetition of a certain cycle (i.e., there exists $T$ such that $v_i+T = v_i$, for any $i \geq i_0$). This decomposition is unique, and we say the “first-touch-vertex” of $v_0$ is $v_{i0}$.

Proof of Observation 2: Since the graph is finite, then some vertices must appear at least twice in $W(v_0)$. Among all such vertices, suppose $u$ is the one that appears the earliest in the walk $W(v_0)$, and the first two appearances are $v_{i0} = u$ and $v_{i1} = u$ and $i_0 < i_1$. Denote $T = i_1 - i_0$. Then it is easy to show $W_2(v_0)$ is the repetitions of the cycle consisting of vertices $v_{i0}, v_{i0+1}, \ldots, v_{i1-1}$, and $W_1(v_0)$ is a directed path from $v_0$ to $v_{i0}$.

The first-touch-vertex $u = u_{i0}$ has the following properties: (i) $u \in C_k$ for some $k$; (ii) there exists a path from $v$ to $u$; (iii) any paths from $v$ to any vertex in the cycle $C_k$ other than $u$ must pass $u$. Note that if $u$ is in some cycle, then its first-touch-vertex is $u$ itself.

As a corollary of Observation 2, there is at least one cycle. Suppose all cycles of $G$ are $C_1, C_2, \ldots, C_K$. Because the outdegree of each vertex is 1, these cycles must be disjoint, i.e., $V(C_i) \cap V(C_j) = \emptyset$ and $A(C_i) \cap A(C_j) = \emptyset$, for any $i \neq j$. Denote the set of vertices in the cycles as

$$V_c = \bigcup_{k=1}^K V(C_1) \cup \cdots \cup V(C_K).$$

Let $u_1, \ldots, u_M$ be the vertices of $C_1, \ldots, C_m$ with indegree at least 2.

Based on Observation 2, starting from any vertex outside $V_c$ there is a unique path that reaches $V_c$. Combining all vertices that reach the cycles at $u_m$ (denoted as $V_m$), and the paths from these vertices to $u_m$, we obtain a directed subgraph $T_m$, which is connected with $V_c$ only via the vertex $u_m$. The subgraphs $T_m$’s are disjoint from each other since they are connected with $V_c$ via different vertices. In addition, each vertex outside of $V_c$ lies in exactly one of the subgraph $T_m$. Thus, we can partition the whole graph into the union of the cycles $C_1, \ldots, C_K$ and the subgraphs $T_1, \ldots, T_M$.

We then show $T_m$’s are trees. For any vertex $v_0$ in the subgraph $T_m$, consider the walk $W(v_0)$. Any path starting from $v_0$ must be a part of $W(v_0)$. Starting from $v_0$ there is only one path from $v_0$ to $u_m$ which is $W_1(v_0)$, according to Observation 2. Therefore, by the definition of a directed tree, $T_m$ is a directed tree with the root $u_m$. Therefore, we can partition the whole graph into the union of the cycles $C_1, \ldots, C_K$ and subgraphs $T_1, \ldots, T_M$ with disjoint edge sets; in addition, the edge sets of the cycles are disjoint, and the root of $T_m$ must be in certain cycle $C_k$. It is easy to verify the properties stated in Lemma 1. This finishes the proof.

**J.2.2 Proof of Claim 11**

We first prove the case for $d \geq 2$. Suppose the corresponding graph for $Y$ is $G$, and $G$ is decomposed into the union of cycles $C_1, \ldots, C_K$ and trees $T_1, \ldots, T_m$. We perform the following operation: pick an arbitrary tree $T_m$ with the root $u_m$. The tree is non-empty, thus there must be an edge $e$ with the head $u_m$.

Suppose $v$ is the tail of the edge $e$. Now we remove the edge $e = (v, u_m)$ and create a new edge $e' = (v, v)$. The new edge corresponds to $y_v = x_v$. The old edge $(v, u_m)$ corresponds to
\( y_v = x_{u_m} \) (and a term \( h(f(x_{u_m}) - f(x_v)) \)) if \( u_m \leq n \) or \( y_v = y_{u_{m-n}} \neq \{x_1, \ldots, x_n\} \) (and a term \( h(f(y_{u_{m-n}}) - f(x_v)) \)) if \( u_m > n \). This change corresponds to the change of \( y_v \): we change \( y_v = x_{u_m} \) (if \( u_m \leq n \)) or \( y_v = y_{u_{m-n}} \) (if \( u_m > n \)) to \( y_v = x_v \). Let \( y_i = y_i \) for any \( i \neq v \), and \( \hat{Y} = (\hat{y}_1, \ldots, \hat{y}_n) \) is the new point.

Previously \( v \) is in a tree \( T_m \) (not its root), now \( v \) is the root of a new tree, and also part of the new cycle (self-loop) \( C_{K+1} = (v, v') \). In this new graph, the number of vertices in cycles increases by 1, thus the value of \( g \) increases by \(-\frac{1}{n} \log 2\), i.e., \( g(\hat{Y}) - g(Y) = \frac{1}{n} \log 2 \).

Since \( d \geq 2 \), we can find a path in \( \mathbb{R}^d \) from a point to another point without passing any of the points in \( \{x_1, \ldots, x_n\} \). In the continuous process of moving \( y_v \) to \( \hat{y}_v \), the function value will not change except at the end that \( y_v = x_v \). Thus there is a non-increasing path from \( Y \) to \( Y' \), in the sense that along this path the function value of \( g \) does not decrease.

The illustration of this proof is given below.

![Original graph](image1)

(a) Original graph

![Modified graph, with improved function value](image2)

(b) Modified graph, with improved function value

Figure 26: Illustration of the proof of Claim J.1. For the figure on the left, we pick an arbitrary tree with the head being vertex 0, which corresponds to \( y_0 = y_7 \). We change \( y_7 \) to \( \hat{y}_7 = x_7 \) to obtain the figure on the right. Since one more cycle is created, the function value increases by at least \( \frac{1}{n} \log 2 \). This path is a non-decreasing path, thus the claim is also proved.

For the case \( d = 1 \), the above proof does not work. The reason is that the path from \( y_v \) to \( \hat{y}_v \) may touch other points in \( \{x_1, \ldots, x_n\} \) and thus may change the value of \( g \). We only need to make a small modification: we move \( y_v \) in \( \mathbb{R} \) until it touches a certain \( x_i \), which corresponds to a vertex in the tree \( T_m \), at which point a cycle is created, and the function value increases by at least \( \frac{1}{n} \log 2 \). This path is a non-decreasing path, thus the claim is also proved.

### J.3 Proof of Theorem 2

Obviously, \( g(Y) = \phi_R(Y, X) = \frac{1}{n} \sup_{f \in C(\mathbb{R}^d)} \sum_{i=1}^{n} [h(f(x_i) - f(y_i))] = h(0) \) (by picking \( f = 0 \)).

**Step 1: achieving optimal** \( g(Y) \). We prove if \( \{y_1, \ldots, y_n\} = \{x_1, \ldots, x_n\} \), then \( g(Y) = h(0) \).

**Claim J.2.** Assume \( h \) is concave. Then the function \( \xi_R(m) = \sup_{(t_1, \ldots, t_m) \in ZO(m)} \sum_{i=1}^{m} h(t_i) \) satisfies \( \xi_R(m) = mh(0) \), where the set \( ZO(m) = \{t_1, t_2, \ldots, t_m \in \mathbb{R} : \sum_{i=1}^{m} t_i = 0\} \).

The proof of this claim is obvious and skipped here. When \( \{y_1, \ldots, y_n\} = \{x_1, \ldots, x_n\} \), we can divide \([n]\) into multiple cycles \( C_1 \cup \cdots \cup C_K \), each with length \( m_k \), and obtain \( \phi_R(Y, X) = \frac{1}{n} \sup_{f \in C(\mathbb{R}^d)} \sum_{k=1}^{K} m_k h(0) = \frac{1}{n} \sum_{k=1}^{K} k \xi_R(m_k) = \frac{1}{n} \sum_{k=1}^{K} m_k h(0) = h(0) \).

**Step 2: compute** \( g(Y) \) **when** \( y_i \in \{x_1, \ldots, x_n\} \), \( \forall i \). Assume \( y_i \in \{x_1, \ldots, x_n\} \), \( \forall i \). We build a directed graph \( G = (V, A) \) as follows (the same graph as in Appendix [1]). The set of vertices \( V = \{1, 2, \ldots, n\} \) represents \( x_1, x_2, \ldots, x_n \). We draw a directed edge \((i, j) \in A \) if \( y_j = x_j \). Note that it is possible to have a self-loop \((i, i) \), which corresponds to the case \( y_i = x_i \).

According to Lemma 4 this graph can be decomposed into cycles \( C_1, C_2, \ldots, C_K \) and subtrees \( T_1, T_2, \ldots, T_M \). We claim that

\[
\phi_R(Y, X) = \frac{1}{n} \sum_{k=1}^{K} |V(C_k)| h(0) \geq h(0). \tag{31}
\]

The proof of the relation in Eq. (31) is similar to the proof of Eq. (22) used in the proof of Theorem 2 and briefly explained below. One major part of the proof is to show that the contribution of the nodes
in the cycles is $\sum_{k=1}^{K} |V(C_k)|h(0)$. This is similar to Step 1, and is based on Claim J.2. Another major part of the proof is to show that the contribution of the nodes in the subtrees is zero, similar to the proof of Eq. (28). This is because we can utilize Assumption 4.4 to construct a sequence of $f$ values (similar to Eq. (26)) so that

$$f(y_i) - f(x_i) = \begin{cases} 0, & i \in \bigcup_{k=1}^{K} V(C_k), \\ \alpha_N, & i \in \bigcup_{m=1}^{M} V(T_m). \end{cases}$$ (32)

Here $\{\alpha_N\}_{N=1}^{\infty}$ is a sequence of real numbers so that $\lim_{N \to \infty} h(\alpha_N) = \sup \ h(t) = 0$. In the case that $h(\infty) = 0$ like RS-GAN, we pick $\alpha_N = N$. In the case that $h(a) = 0$ for a certain finite number $a$, we can just pick $\alpha_N = a$, $\forall N$ (thus we do not need a sequence but just one choice).

Since the expression of $\phi_R(Y, X)$ in Eq. (31) is a scaled version of the expression of $\phi_{RS}(Y, X)$ (scale by $-\log \frac{2}{h(0)}$), the rest of the proof is the same as the proof of Theorem 2.

**Step 3: function value for general $Y$ and GMR.** This step is the same as the proof of Theorem 1. For the value of general $Y$, we build an “augmented graph” and apply the result in Step 2 to obtain $g(Y)$. To prove GMR, the same construction as the proof of Theorem 1 suffices.

**K Results in Parameter Space**

We will first state the technical assumptions and then present the formal results in parameter space. The results become somewhat technical due to the complication of neural-nets. Suppose the discriminator neural net is $f_\theta$ where $\theta \in \mathbb{R}^f$ and the generator net is $G_w$ where $w \in \mathbb{R}^K$.

**Assumption K.1.** (representation power of discriminator net): For any distinct vectors $v_1, \ldots, v_{2n} \in \mathbb{R}^d$, any $b_1, \ldots, b_{2n} \in \mathbb{R}$, there exists $\theta \in \mathbb{R}^f$ such that $f_\theta(v_i) = b_i$, $i = 1, \ldots, 2n$.

**Assumption K.2.** (representation power of generator net in $W$) For any distinct $z_1, \ldots, z_n \in \mathbb{R}^d$, and any $y_1, \ldots, y_n \in \mathbb{R}^d$, there exists $w \in W$ such that $G_w(z_i) = y_i$, $i = 1, \ldots, n$.

For any given $Z = (z_1, \ldots, z_n) \in \mathbb{R}^{d \times n}$, and any $w \in W \subseteq \mathbb{R}^K$, we define a set $G^{-1}(Y; Z)$ as follows: $w \in G^{-1}(Y; Z)$ iff $G_w(Z) = Y$ and $w \in W$.

**Assumption K.3.** (path-keeping property of generator net; duplication of Assumption 4.6): For any distinct $z_1, \ldots, z_n \in \mathbb{R}^d$, the following holds: for any continuous path $Y(t)$, $t \in [0, 1]$ in the space $\mathbb{R}^{d \times n}$ and any $w_0 \in G^{-1}(Y(0); Z)$, there is a continuous path $w(t)$, $t \in [0, 1]$ such that $w(0) = w_0$ and $Y(t) = G_w(t)(Z)$, $t \in [0, 1]$.

We will present sufficient conditions for these assumptions later. Next we present two main results on the landscape of GANs in the parameter space.

**Proposition K.1.** (formal version of Prop. 1) Consider the separable-GAN problem $\min_{w \in \mathbb{R}^K} \varphi_{sep}(w)$, where $\varphi_{sep}(w) = \sup_\theta \frac{1}{n} \sum_{i=1}^{n} \left[ h_1(f_\theta(x_i)) + h_2(-f_\theta(G_w(z_i))) \right]$. Suppose $h_1, h_2$ satisfy the same assumptions of Theorem 1. Suppose $G_w$ satisfies Assumption K.2 and Assumption 4.6 (with certain $W$). Suppose $f_\theta$ satisfies Assumption K.1. Then there exists at least $(n^2 - n)!$ distinct $w \in W$ that are not global-min-reachable.

**Proposition K.2.** (formal version of Prop. 2) Consider the RpGAN problem $\min_{w \in \mathbb{R}^K} \varphi_R(w)$, where $\varphi_R(w) = \sup_\theta \frac{1}{n} \sum_{i=1}^{n} \left[ h(f_\theta(x_i)) - f_\theta(G_w(z_i)) \right]$. Suppose $h$ satisfies the same assumptions of Theorem 2. Suppose $G_w$ satisfies Assumption K.2 and Assumption 4.6 (with certain $W$). Suppose $f_\theta$ satisfies Assumption K.1. Then any $w \in W$ is global-min-reachable for $\varphi_R(w)$.

We have presented two generic results that relies on a few properties of the neural-nets. These properties can be satisfied by certain neural-nets, as discussed next. Our results largely rely on recent advanced in neural-net optimization theory.

**K.1 Sufficient Conditions for the Assumptions**

In this part, we present a set of conditions on neural nets that ensure the assumptions to hold. We will discuss more conditions in the next subsection.

**Assumption K.4.** (mildly wide) The last hidden layer has at least $\bar{n}$ neurons, where $\bar{n}$ is the number of input vectors.
The assumption of width is common in recent theoretical works in neural net optimization (e.g. [50][73][2]). For the generator network, we set \( n = \bar{n} \); for the discriminator network, we set \( \bar{n} = 2n \).

**Assumption K.5.** *(smooth enough activation)* The activation function \( \sigma \) is an analytic function, and the \( k \)-th order derivatives \( \sigma^{(k)}(0) \) are non-zero, for \( k = 0, 1, 2, \ldots, \bar{n} \), where \( \bar{n} \) is the number of input vectors.

The assumption of the neuron activation is satisfied by sigmoid, tanh, SoftPlus, swish, etc.

For the generator network, consider a fully neural network \( G_w(z) = W_H \sigma(W_{H-1} \ldots W_2 \sigma(W_1 z)) \) that maps \( z \in \mathbb{R}^d \) to \( G_w(z) \in \mathbb{R}^d \). Define \( T_k(z) = \sigma(W_{k-1} \ldots W_2 \sigma(W_1 z)) \in \mathbb{R}^{d_k} \) where \( d_k \) is the number of neurons in the \( k \)-th hidden layer. Then we can write \( G_w(z) = W_H T_H(z) \), where \( W_H \in \mathbb{R}^{d \times d_H} \). Let \( Z = (z_1, \ldots, z_n) \) and let \( T_k(Z) = (T_k(z_1), \ldots, T_k(z_n)) \in \mathbb{R}^{d_k \times n} \), \( k = 1, 2, \ldots, H \). Define \( W = \{ w = (W_1, \ldots, W_H) : T_H(Z) \text{is full rank} \} \).

We will prove that under these two assumptions on the neural nets, the landscape of RpGAN is better than that of SepGAN.

**Proposition K.3.** Suppose \( h_1, h_2, h \) satisfy assumptions in Theorem 1 and Theorem 2. Suppose \( G_w, f_0 \) satisfies Assum. K.3 and K.2 (\( \bar{n} = n \) for \( G_w \), and \( \bar{n} = 2n \) for \( f_0 \)). Then there exist at least \((n^n - n!)\) distinct \( w \in W \) that are not GMR for \( \phi_{\text{Sep}}(w) \). In contrast, any \( w \in W \) is global-min-reachable for \( \phi_{\text{Hi}}(w) \).

This proposition is the corollary of Prop. K.1 and Prop. K.2. We only need to verify the assumptions in the two propositions. The following series of claims provide such verification.

**Claim K.1.** Suppose Assumptions K.3 and K.5 hold for the generator net \( G_w \) with distinct input \( z_1, \ldots, z_n \). Then \( W = \{ (W_1, \ldots, W_H) : T_H(Z) \text{is full rank} \} \) is a dense set in \( \mathbb{R}^H \). In addition, Assumption K.2 holds.

This full-rank condition was used in a few works of neural-net landscape analysis (e.g. [72]). In GAN area, [7] studied invertible generator nets where the weights are restricted to a subset of \( \mathbb{R}^K \) to avoid singularities. As the set \( W \) is dense, intuitively the iterates will stay in this set for most of the time. However, rigorously proving that the iterates stay in this set is not easy, and is one of the major challenges of current neural-network analysis. For instance, [38] shows that for very wide neural networks with proper initialization along the training trajectory of gradient descent the neural-tangent kernel (a matrix related to \( T_H(Z) \)) is full rank. A similar analysis can prove that the matrix \( T_H(Z) \) stays full rank during training under similar conditions. We do not attempt to develop the more complicated convergence analysis for general neural-nets here and leave it to future work.

**Claim K.2.** Suppose Assumptions K.3 and K.5 hold for the generator net \( G_w \) with distinct input \( z_1, \ldots, z_n \). Then it satisfies Assumption 4.6 with \( W \) defined in Claim K.1.

Assumption K.1 can be shown to hold under a similar condition to that in Claim K.1.

**Claim K.3.** Consider a fully connected neural network \( f_0(z) = \theta_H \sigma(\theta_{H-1} \ldots \theta_2 \sigma(\theta_1 z)) \) that maps \( u \in \mathbb{R}^d \) to \( f_0(u) \in \mathbb{R}^d \) and suppose Assumptions K.3 and K.5 hold. Then Assumption K.1 holds.

The proofs of the claims are given in Appendix K.5.

With these claims, we can immediately prove Prop. K.3.

**Proof of Prop. K.3** According to Claim K.2, K.1, K.3 the assumptions of Prop. K.3 imply the assumptions of Prop. K.1 and Prop. K.2. Therefore, the conclusions of Prop. K.1 and Prop. K.2 hold. Since the conclusion of Prop. K.3 is the combination of the the conclusions of Prop. K.1 and Prop. K.2 it also holds.

**K.2 Other Sufficient Conditions**

Assumption K.3 (path-keeping property) is the key assumption. Various results in neural-net theory can ensure this assumption (or its variant) holds, and we have utilized one of the simplest such results in the last subsection. We recommend to check [80] which describes a bigger picture about various landscape results. In this subsection, we briefly discuss other possible results applicable to GAN.

We start with a strong conjecture about neural net landscape, which only requires a wide final hidden layer but no condition on the depth and activation.
**Conjecture K.1.** Suppose $g_0$ is a fully connected neural net with any depth and any continuous activation, and it satisfies Assumption K.4 (i.e. a mildly wide final hidden layer). Assume $f(y, \hat{y})$ is convex in $\hat{y}0$, then the empirical loss function of a supervised learning problem $\sum_{i=1}^{n} f(y_i, g_0(x_i))$ is global-min-reachable for any point.

We then describe a related conjecture for GAN, which is easy to prove if Conjecture K.1 holds.

**Conjecture 1** (informal): Suppose $G_w$ is a fully connected net satisfying Assump. K.4 (i.e. a mildly wide final hidden layer). Suppose $G_w$ and $f_0$ are expressive enough (i.e. Assump. K.2 and Assump. K.1 hold). Then the RpGAN loss has a benign landscape, in the sense that any point is GMR for $\varphi_R(w)$. In contrast, the SepGAN loss does not have this property.

Unfortunately, we are not aware of any existing work that has proved Conjecture K.1, thus we are not able to prove Conjecture 1 above for now. Venturi et al. [84] proved a special case of Conjecture K.1 for $L = 1$ (one hidden layer), and other works such as Li et al. [50] prove a weaker version of Conjecture K.1 see [80] for other related results. The precise version of Conjecture K.1 seems non-trivial to prove.

We list two results on GAN that can be derived from weaker versions of Conjecture K.1; both results apply to the whole space instead of the dense subset $W$.

**Result 1** (1-hidden-layer): Suppose $G_w$ is a 1-hidden-layer network with any continuous activation and any number of layers. Suppose it satisfies Assump. K.4 (i.e. a mildly wide final hidden layer). Suppose $G_w$ and $f_0$ are expressive enough (i.e. Assump. K.2 and Assump. K.1 hold). Then the RpGAN loss satisfies GMR for any point. This result is based on Venturi et al. [84].

**Result 2**: Suppose $G_w$ is a fully connected network with any continuous activation and any number of layers. Suppose it satisfies Assump. K.4 (i.e. a mildly wide final hidden layer). Suppose $G_w$ and $f_0$ are expressive enough (i.e. Assump. K.2 and Assump. K.1 hold). Then the RpGAN loss has no sub-optimal set-wise local minima (see [50, Def. 1] for the definition). This result is based on Li et al. [50].

Due to space constraint, we do not present the proofs of the above two results (combining them with GANs is somewhat cumbersome). The high-level proof framework is similar to that of Prop. K.3.

**K.3 Proofs of Propositions for Parameter Space**

**Proof of Proposition K.1.** The basic idea is to build a relation between the points in the parameter space to the points in the function space.

Denote $L_{sep}(w; \theta) = \frac{1}{n} \sum_{i=1}^{n} [h_1(f_0(x_i)) + h_2(-f_0(G_w(z_i)))]$, then $\varphi_{sep}(w) = \sup_{\theta} L_{sep}(w; \theta)$. Denote $L_{sep}(Y; f) = \frac{1}{2} \sum_{i=1}^{n} [h_1(f(x_i)) + h_2(-f(y_i))]$, and $\phi(Y, X) = \sup_{f} L_{sep}(Y; f)$. Note that in the definition of the two functions above, the discriminator is hidden in the $\sup$ operators, thus we have freedom to pick the discriminator values (unlike the generator space which we have to check all $w$ in the inverse of $Y$).

Our goal is to analyze the landscape of $\varphi_{sep}(w)$, based on the previously proved result on the landscape of $\phi(Y, X)$. We first show that the image of $\varphi_{sep}(\hat{w})$ is the same as that of $\phi_{sep}(\hat{Y}, X)$.

Define $G^{-1}(Y) \triangleq \{w : G_w(z_i) = y_i, i = 1, \ldots, n\}$. We first prove that

$$\phi_{sep}(\hat{Y}, X) = \varphi_{sep}(\hat{w}), \forall \hat{w} \in G^{-1}(\hat{Y}).$$

Suppose $\phi_{sep}(\hat{Y}, X) = \alpha$. This implies that $L_{sep}(\hat{Y}; f) \leq \alpha$ for any $f$; in addition, for any $\epsilon > 0$ there exists $\hat{f} \in C(\mathbb{R}^d)$ such that

$$L_{sep}(\hat{Y}; \hat{f}) \geq \alpha - \epsilon.$$  \hspace{1cm} (34)

According to Assumption K.1, there exists $\theta^*$ such that $f_{\theta^*}(x_i) = \hat{f}(x_i)$, $\forall i$, and $f_{\theta^*}(u) = \hat{f}(u), \forall u \in \{y_1, \ldots, y_n\} \setminus \{x_1, \ldots, x_n\}$. In other words, there exists $\theta^*$ such that

$$f_{\theta^*}(x_i) = \hat{f}(x_i), f_{\theta^*}(y_i) = \hat{f}(y_i), \forall i.$$  \hspace{1cm} (35)
Then we have
\[ L_{\text{sep}}(\bar{w}; \theta^*(\epsilon)) = \frac{1}{2m} \sum_{i=1}^{n} [h_1(f_\theta(x_i)) + h_2(-f_\theta(G_{\bar{w}}(z_i)))] = \frac{1}{2m} \sum_{i=1}^{n} [h_1(f_\theta(x_i)) + h_2(-f_\theta(\bar{y}_j))] \]
\[ = \frac{1}{2m} \sum_{i=1}^{n} [h_1(\bar{f}(x_i)) + h_2(-\bar{f}(\bar{y}_j))] = L_{\text{sep}}(\bar{Y}; \hat{f}) \geq \alpha - \epsilon. \]

In the above chain, (i) is due to the assumption \( \bar{w} \in G^{-1}(\hat{Y}) \) (which implies \( G_{\bar{w}}(z_j) = \bar{y}_j \)), (ii) is due to the choice of \( \theta^* \). (iii) is due to (34).

Therefore, we have \( \varphi_{\text{sep}}(\bar{w}) = \sup_\theta L_{\text{sep}}(\bar{w}; \theta) \geq L_{\text{sep}}(\bar{w}; \theta^*(\epsilon)) \geq \alpha - \epsilon \). Since this holds for any \( \epsilon \), we have \( \varphi_{\text{sep}}(\bar{w}) \geq \alpha \). Similarly, from \( L_{\text{sep}}(\bar{w}; \theta) \leq \alpha \) we can obtain \( \varphi_{\text{sep}}(\bar{w}) \leq \alpha \). Therefore \( \varphi_{\text{sep}}(\bar{w}) = \alpha = \varphi_{\text{sep}}(\hat{Y}, X) \). This finishes the proof of (33).

Define
\[ Q(X) \triangleq \{ Y = (y_1, \ldots, y_n) \mid y_i \in \{x_1, \ldots, x_n\}, i \in \{1, 2, \ldots, n\}; y_i = y_j \text{ for some } i \neq j \}. \]

Any \( Y \in Q(X) \) is a mode-collapsed pattern. According to Theorem [1], any \( Y \in Q(X) \) is a strict local minimum of \( \varphi_{\text{sep}}(Y, X) \), and thus \( Y \) is not GMR. Therefore \( \bar{w} \in G^{-1}(Y) \) where \( Y \) is not GMR; this is because a non-decreasing path in the parameter space will be mapped to a non-decreasing path in the function space, causing contradiction. Finally, according to Assumption K.2, for any \( Y \) there exists at least one pre-image \( w \in G^{-1}(Y) \cap W \). There are \( (n^n - n!) \) elements in \( Q(X) \), thus there are at least \( (n^n - n!) \) points in \( W \) that are not global-min-reachable. \( \square \)

**Proof of Proposition K.3** Similar to Eq. (33), we have \( \varphi_R(w) = \varphi_R(Y, X) \) for any \( w \in G^{-1}(Y) \). We need to prove that there is a non-decreasing path from any \( w_0 \in W \) to \( w^* \), where \( w^* \) is a certain global minimum. Let \( Y_0 = G_{\varphi_R}(z_1, \ldots, z_n) \). According to Thm. [2] there is a continuous path \( Y(t) \) from \( Y_0 \) to \( Y^* \) along which the loss value \( \varphi_R(Y(t), X) \) is non-increasing. According to Assum. K.6, there is a continuous path \( w(t) \) such that \( w(0) = \bar{w}, Y(t) = G_{\varphi_R}(Z), t \in [0, 1] \). Along this path, the value \( \varphi_R(w(t)) = \varphi_R(Y(t), X) \) is non-increasing, and at the end the function value \( \varphi_R(w(1)) = \varphi_R(Y^*, X) \) is the minimal value of \( \varphi_R(w) \). Thus the existence of such a path is proved. \( \square \)

**K.4 A technical lemma**

We present a technical lemma, that slightly generalizes [50] Proposition 1.

**Assumption K.6** \( v_1, v_2, \ldots, v_m \in \mathbb{R}^d \) are distinct, i.e., \( v_i \neq v_j \) for any \( i \neq j \).

**Lemma 2.** Define \( T_H(V) = (\sigma(W_{H-1} \ldots W_2 \sigma(W_1 v_i)))_{i=1}^{m} \in \mathbb{R}^{dh \times m} \). Suppose Assumptions K.3 and K.6 hold. Then the set \( \Omega = \{(W_1, \ldots, W_{H-1}) : \text{rank}(T_H(V)) < m\} \) has zero measure.

This claim is slightly different from [50] Proposition 1, which requires the input vectors to have one distinct dimension (i.e., there exists \( j \) such that \( v_{1j}, \ldots, v_{mj} \)) are distinct; here we only require the input vectors to be distinct. It is not hard to link “distinct vectors” to “vectors with one distinct dimension” by a variable transformation.

**Claim K.4.** Suppose \( v_1, \ldots, v_m \in \mathbb{R}^d \) are distinct. Then for generic matrix \( W \in \mathbb{R}^{d \times d} \), for the vectors \( \bar{v}_i = W v_i \in \mathbb{R}^d \), \( i = 1, \ldots, n \), there exists \( j \) such that \( \bar{v}_{1j}, \ldots, \bar{v}_{mj} \) are distinct.

**Proof.** Define the set \( \Omega_0 = \{ u \mid u \in \mathbb{R}^{1 \times d}, \exists i \neq j \text{ s.t. } u^T v_i = u^T v_j \}. \) This is the union of \( d(d-1) \) hyperplanes \( \Omega_{ij} = \{ u \mid u \in \mathbb{R}^{1 \times d}, u^T v_i = u^T v_j \}. \) Each hyperplane \( \Omega_{ij} \) is a zero-measure set, thus the union of them \( \Omega_0 \) is also a zero-measure set. Let \( u \) be the first row of \( W \), then \( u \) is generic vector and thus not in \( \Omega_0 \), which implies \( \bar{v}_{1j}, \ldots, \bar{v}_{mj} \) are distinct. \( \square \)

**Proof of Lemma 2** Pick a generic matrix \( A \in \mathbb{R}^{d \times d} \), then \( \bar{v}_i = A v_i \in \mathbb{R}^{d \times d} \) has one distinct dimension, i.e., there exists \( j \) such that \( \bar{v}_{1j}, \ldots, \bar{v}_{mj} \) are distinct. In addition, we can assume \( A \) is full rank (since it is generic). Define
\[ T_{\bar{H}}(V) = (\sigma(W_{H-1} \ldots W_2 \sigma(W_1 \bar{v}_i)))_{i=1}^{m} \in \mathbb{R}^{dh \times m}. \]
According to [50] Prop. 1, the set \( \hat{\Omega} = \{(W_1, W_2, W_3, \ldots, W_{H-1}) : \text{rank}(T_{\bar{H}}(V)) < m\} \) has zero measure. With the transformation \( \eta_0(W_1) = W_1 A^{-1} \), we have \( \sigma(W_{H-1} \ldots W_2 \sigma(W_1 \bar{v}_i)) = \]
W-GAN is a popular formulation of GAN, so a natural question is whether we can prove a similar landscape result for W-GAN. Consider W-GAN formulation (empirical version) \( \min_Y \phi_W(Y, X) \), where
\[
\phi_W(Y, X) = \max_{|f|_\infty \leq 1} \frac{1}{n} \sum_{i=1}^{n} |f(x_i) - f(y_i)|.
\]
For simplicity we consider the same number of generated samples and true samples. It can be viewed as a special case of RpGAN where \( h(t) = -t \); it can also be viewed as a special case of SepGAN where \( h_1(t) = h_2(t) = -t \).

However, the major complication is the Lipschitz constraint. It makes the computation of the function values much harder. For the case of \( n = 2 \), the function value of \( \phi_W(Y, X) \) is provided in the following claim.

**Claim L.1.** Suppose \( n = 2 \). Denote \( a_1 = x_1, a_2 = x_2, a_3 = y_1, a_4 = y_2 \). The value of \( \phi_W(Y, X) \) is
\[
\max_{u_1, u_2, u_3, u_4 \in \mathbb{R}} u_1 + u_2 - u_3 - u_4,\]
\[
s.t. \quad |u_i - u_j| \leq |a_i - a_j|, \forall i, j \in \{1, 2, 3, 4\}.
\]

This claim is not hard to prove, and we skip the proof here.

This claim indicates that computing \( \phi_W(Y, X) \) is equivalent to solving a linear program (LP). Solving LP itself is computationally feasible, but our landscape analysis requires to infer about the global landscape of \( \phi_W(Y, X) \) as a function of \( Y \). In classical optimization, it is possible to state that the optimal value of an LP is a convex function of a certain parameter (e.g. the coefficient of the objective). But in our LP \( y_i \)'s appear in multiple positions of the LP, and we are not aware of an existing result that can be readily applied.

Similar to Kantorovich-Rubinstein Duality, we can write down the dual problem of the LP where the objective is linear combination of \( |a_i - a_j| \). However, it is still not clear what to say about the global landscape, due to the lack of closed-form solutions.

Finally, we remark that although W-GAN has a strong theoretical appeal, it did not replace JS-GAN or simple variants of JS-GAN in recent GAN models. For instance, SN-GAN [67] and BigGAN [18] use hinge-GAN.
Table 7: CNN models for CIFAR-10 and STL-10 used in our experiments on high resolution image generation. All layers of D use ReLU-0.1 (except the final dense "linear" layer).

| Layer | Activation | Stride
|-------|------------|--------|
| ResBlock up 256/c | ResBlock down 128/c |
| ResBlock up 256/c | ResBlock down 128/c |
| ResBlock up 256/c | ResBlock down 128/c |
| BN, ReLU, 3 × 3 conv, 3 Tanh | Global sum pooling |
| dense = 1 |

Table 8: CNN model architecture for size 256 LSUN used in our experiments on high resolution image generation. All layers of G use ReLU (except one layer with Tanh); all layers of D use LReLU-0.1.

| Layer | Activation | Stride
|-------|------------|--------|
| ResBlock up 256/c | ResBlock down 128/c |
| ResBlock up 256/c | ResBlock down 128/c |
| ResBlock up 256/c | ResBlock down 128/c |
| BN, ReLU, 3 × 3 conv, 3 Tanh | Global sum pooling |
| dense = 1 |

Table 9: Resnet architecture for CIFAR-10, co=1, 2 and 4 for the regular, 1/2 and 1/4 channel structures respectively.

| Layer | Activation | Stride
|-------|------------|--------|
| ResBlock up 256/c | ResBlock down 128/c |
| ResBlock up 256/c | ResBlock down 128/c |
| ResBlock up 256/c | ResBlock down 128/c |
| BN, ReLU, 3 × 3 conv, 3 Tanh | Global sum pooling |
| dense -> 1 |

Table 10: Resnet architecture for STL-10, co=1, 2 and 4 for the regular, 1/2 and 1/4 channel structures respectively.

| Layer | Activation | Stride
|-------|------------|--------|
| ResBlock up 256/c | ResBlock down 128/c |
| ResBlock up 256/c | ResBlock down 128/c |
| ResBlock up 256/c | ResBlock down 128/c |
| BN, ReLU, 3 × 3 conv, 3 Tanh | Global sum pooling |
| dense -> 1 |

Table 11: BottleNeck Resnet models for CIFAR-10. BRes refers to BottleNeck ResBlock. BRes (a, b, c) refers to the Bottleneck resblock with (input, hidden and output) being (a, b, c).

| Layer | Activation | Stride
|-------|------------|--------|
| ResBlock up 256/c | ResBlock down 128/c |
| ResBlock up 256/c | ResBlock down 128/c |
| ResBlock up 256/c | ResBlock down 128/c |
| BN, ReLU, 3 × 3 conv, 3 Tanh | Global sum pooling |
| dense -> 1 |

Table 12: BottleNeck Resnet models for STL-10.
Figure 16: Generated CIFAR-10 samples with CNN.
Figure 17: Generated CIFAR-10 samples on ResNet.
Figure 18: Generated STL-10 samples with CNN.
Figure 19: Generated STL-10 samples with ResNet.
Figure 20: Generated 256 × 256 Church and Tower Image by JS-GAN and RS-GAN.