Stabilizing GANs Under Limited Resources via Dynamic Machine Ordering

by

Joshua Caseiro DeOliveira

A Thesis

Submitted to the Faculty

of the

WORCESTER POLYTECHNIC INSTITUTE

In partial fulfillment of the requirements for the

Degree of Master of Science

in

Data Science

September 2023

APPROVED:

_____

Professor Elke A. Rundensteiner, Thesis Advisor

_____

Professor Oren Mangoubi, Thesis Reader

ABSTRACT

Generative Adversarial Networks (GANs) are a generative framework with a notorious reputation for instability. Despite significant work in attempting to improve stability, training remains extremely difficult in practice. Nearly all GAN optimization methods are built on either simultaneous (Sim-GDA) or alternating (Alt-GDA) gradient descent-ascent, where the generator and discriminator are updated either at the same time iteratively or in a fixed pattern. In this paper, we prove for simple GANs, for which training had been proven non-convergent under Sim-GDA and Alt-GDA, that our newly introduced training method is Lyapunov-stable. We then design a novel oracle-guided GDA training strategy called Dynamic-GDA that leverages generalized analogs of the properties exhibited in the simple case. We also prove that in contrast to Sim/Alt-GDA, GANs with Dynamic-GDA achieve Lyapunov-stable training with non-infinitesimal learning rates. Empirically, we show Dynamic-GDA improves convergence orthogonally to common stabilizing techniques on 8 classes of GAN models and 7 different data sets.

1 INTRODUCTION

Generative Adversarial Networks (GANs) Goodfellow et al. (2014), a class of deep generative models, have been adopted for numerous applications in domains such as image augmentation Zhao et al. (2020), cyber-security Arora & Shantanu (2022), and imitation learning Ho & Ermon (2016). GANs pose the learning of the data distribution as a min-max problem, in which the generator aims to synthesize samples that the discriminator cannot distinguish from real data. However, due to the challenging nature of min-max problems Nagarajan & Kolter (2017), GANs are notoriously difficult to train and often do not converge close to the ideal distribution Kodali et al. (2017). To tackle this, many strategies have been explored to stabilize training Gui et al. (2021), including different architectures for specific data modalities Radford et al. (2015), different loss functions Arjovsky et al. (2017), or new regularizations Miyato et al. (2018). However, we suggest that one of the most overlooked aspects of GAN training is the method of optimization itself. Specifically, we pose that the *order* in which the generator and discriminator are updated during training is a promising yet understudied facet of GAN optimization.

Nearly all GAN optimization methods are built on simultaneous gradient descent-ascent (Sim-GDA) Nowozin et al. (2016) or alternating gradient descent-ascent (Alt-GDA) Goodfellow et al.

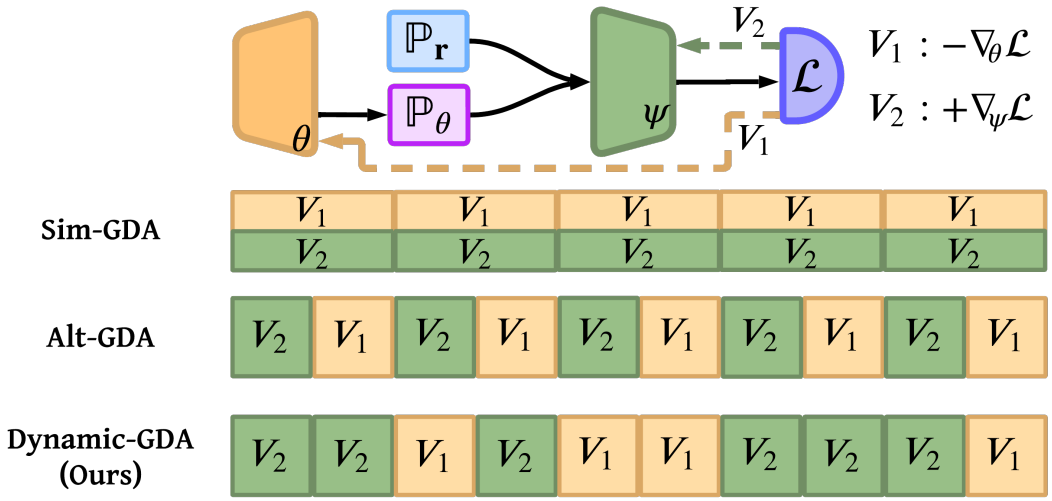

Figure 1: Most GAN optimization relies on Sim-GDA or Alt-GDA for when to update the generator (V_1) and the discriminator (V_2). We propose a new class of optimization that utilizes a dynamic series of updates that can be shown to improve guaranteed stability.

(2014), which update the generator and discriminator either at the same time or in a fixed pattern. Unfortunately, Alt-GDA and Sim-GDA only have proven convergence guarantees under *impractical assumptions* such as unconstrained training time or infinitesimally small learning rates Mescheder et al. (2017); Nagarajan & Kolter (2017); Kodali et al. (2017). Consequently, stabilizing GANs in practice remains an open problem Saxena & Cao (2021).

We tackle this important problem by posing a novel generalization of GAN training for adaptive optimization. Using this generalization, we show theoretically and empirically that for simple GANs equipped with loss functions that were proven non-convergent under Sim-GDA or Alt-GDA, we can construct a novel GDA method that demonstrates highly stable behavior. Further, we prove that simple GANs with this construction have strong Lyapunov stability. We expand upon these insights in order for them to be generally beneficial for pragmatic training. In doing so, we propose a novel optimization method called Dynamic-GDA that can stabilize GANs in general under practical training assumptions. Lastly, we prove Dynamic-GDA is Lyapunov stable.

Dynamic-GDA works by adaptively choosing whether to update the parameters of the generator or discriminator in each iteration of training based on the dynamics of the relationship between the real and synthetic data distributions in the discriminator's projection. This adaptive optimization realizes a dynamic update order - in place of the previous fixed patterns. (See Figure 1).

To show that Dynamic-GDA is a practical method, we conduct a comprehensive study across 7 datasets and 8 classes of GANs to show that it is statistically significant that GANs equipped with Dynamic-GDA stabilize convergence across a wide range of datasets and GAN evaluation metrics.

In summary, the contributions of this work include:

- We theoretically show that state-of-the-art GAN methods equipped with adaptive updating patterns boast stronger stability than when equipped with classical training like Alt-GDA or Sim-GDA.
- We propose a novel optimization algorithm for GAN training called Dynamic-GDA. We theoretically show that Dynamic-GDA guarantees Lyapunov stability.
- We conduct an empirical study to demonstrate state-of-the-art GAN stabilization methods equipped with Dynamic-GDA stabilize convergence better than Alt-GDA and Sim-GDA in a statistically significant way.

We refer the reader to the appendix for *all proofs* discussed throughout the paper.

2 GAN Training Generalized with Oracles

2.1 Classical GAN Training with Alt/Sim-GDA

GANs are a *generative model* in which two continuous, differentiable, and parametric operators, namely a generator $g(\cdot;\theta)$ and discriminator $d(\cdot;\psi)$, utilize the parameters $\theta \in \mathbb{R}^n$ and $\psi \in \mathbb{R}^m$ to optimize a min-max function $\mathcal{L}$:

$$\min_{\theta} \max_{\psi} \mathcal{L}(\theta, \psi) \tag{1}$$

Here, d is tasked with separating real samples from a target distribution $\mathbb{P}_{\mathbf{r}}$ from fake samples from a synthetic distribution $\mathbb{P}_\theta$, while g adversely aims to produce a synthetic distribution $\mathbb{P}_\theta$ that fools d.

An equilibrium is reached, once (θ, ψ) lies on a saddle-point of $\mathcal{L}$, namely, $\nabla_\theta \mathcal{L} = \nabla_\psi \mathcal{L} = 0$ and $\nabla^2_\psi \mathcal{L} < 0 < \nabla^2_\theta \mathcal{L}$. Conceptually, the above implies that the synthetic distribution matches the target distribution, $\mathbb{P}_\theta = \mathbb{P}_{\mathbf{r}}$, and d can no longer reasonably distinguish $\mathbb{P}_\theta$ from $\mathbb{P}_{\mathbf{r}}$ Goodfellow et al. (2014). The original $\mathcal{L}$ used for GAN training Goodfellow et al. (2014) is shown in Equation 2.

$$\mathcal{L} = \mathbb{E}_{x \sim \mathbb{P}_{\mathbf{r}}} \log[d(x;\psi)] + \mathbb{E}_{\hat{x} \sim \mathbb{P}_\theta} \log[1 - d(\hat{x};\psi)] \tag{2}$$

To achieve an equilibrium, conventional GAN optimization utilizes either Sim-GDA Nowozin et al. (2016), where θ and ψ are updated simultaneously each iteration (Alg. 3 in Appendix), or Alt-GDA Goodfellow et al. (2014), where either θ or ψ is updated depending on a fixed schedule (Alg. 4 in Appendix). Nearly all GAN optimization uses either Sim-GDA, Alt-GDA, or their derivatives.

2.2 FORMULATING GAN TRAINING USING DECISION ORACLES

When we train a GAN for a single iteration, we optimize the parameters θ and ψ according to some update field $V(\theta, \psi)$. Given this, we define a set of update fields $\mathbf{V} = \{V_1, V_2, V_3\}$ that can encapsulate the optimizations by Alt-GDA and Sim-GDA Mescheder et al. (2018).

$$\mathbf{V} = \begin{cases} V_1 \rightarrow (-\nabla_\theta \mathcal{L}(\theta, \psi), 0) \\ V_2 \rightarrow (0, \nabla_\psi \mathcal{L}(\theta, \psi)) \\ V_3 \rightarrow (-\nabla_\theta \mathcal{L}(\theta, \psi), \nabla_\psi \mathcal{L}(\theta, \psi)) \end{cases} \tag{3}$$

The elements V_1, V_2 and V_3 of $\mathbf{V}$ in Equation 3 are functional representations of a gradient update applied to the parameters of either solely the generator g (V_1), solely the discriminator d (V_2), or both (V_3) for a single iteration.

In this work, we demonstrate both theoretically and experimentally that adapting the order in which these fields are applied during training improves and in some cases stabilizes GAN optimization. For this, we now generalize training by introducing the notion of a *decision oracle* ζ that determines the update field V for each iteration. Algorithm 1 formulates GAN training according to a prescribed decision oracle. We now can redefine Sim-GDA and Alt-GDA as decision oracles as shown in Equations 4 and 5, where n_d is the number of discriminator updates per generator update.

$$\zeta^{\mathrm{Sim}}(i) = V_3 \tag{4}$$

$$\zeta^{\mathrm{Alt}}(i) = \begin{cases} V_1 : \ i \equiv n_\mathrm{d} \mod (n_\mathrm{d} + 1) \\ V_2 : \ \text{otherwise} \end{cases} \tag{5}$$

Sim-GDA and Alt-GDA choose V using the current iteration count i, while in contrast decision oracles can be any method as long as ζ produces a field $V : (\boldsymbol{\theta}, \boldsymbol{\psi}) \rightarrow (\boldsymbol{\theta}, \boldsymbol{\psi})$.

Algorithm 1 Generalized GAN Training

Require: (θ_0, ψ_0) initial parameters. ζ, decision oracle. η, learning rate.
  while (θ, ψ) not converged **do**
    Get ω, the input required by ζ each iteration
    $V \leftarrow \zeta(\omega)$
    $(\theta, \psi) \leftarrow (\theta, \psi) + \eta V(\theta, \psi)$
  end while

3 BUILDING OUR SOLUTION FOR DIRAC-GANS

We first illustrate our proposed strategy using Dirac-GAN Mescheder et al. (2018), while in Section 4, we generalize this to a wide range of GANs. For Dirac-GAN, we propose a simple yet surprisingly effective decision oracle that boasts much greater stability than using Sim-GDA or Alt-GDA. We prove that for Dirac-GANs, our dynamic dirac-oracle introduced in Equation 7 achieves remarkably tight upper-bounds on the distance that (θ, ψ) will ever be from the saddlepoint when using non-infinitesimally small learning rates: *something not possible for Sim-GDA nor Alt-GDA*.

3.1 DIRAC-GANS

First studied by Mescheder et al. Mescheder et al. (2018), the Dirac-GAN is one of the simplest GAN formulation. In this setting, $\dim(\theta) = \dim(\psi) = 1$, the target distribution is a univariate delta distribution fixed at 0, and the generator samples from a univariate dirac-delta distribution fixed at 1.

$$\begin{aligned} d(x;\psi) &= x\psi, \quad x \sim \delta(0) \\ g(z;\theta) &= z\theta, \quad z \sim \delta(1) \end{aligned} \tag{6}$$

Equation 6 illustrates the architectures of the generator and discriminator as well as real and synthetic distributions. Based on this formulation, exactly one saddle-point (θ^*, ψ^*) exists located at the parameter pair $(0, 0)$.

3.2 DYNAMIC DECISION ORACLE FOR DIRAC-GANS

We show that we achieve superlinear convergence during Dirac-GAN training by using the decision oracle ζ^{Dirac} in Equation 7. ζ^{Dirac} requires the current parameters of the generator and discriminator, (θ, ψ), as input in each iteration.

$$\zeta^{\text{Dirac}}(\theta, \psi) = \begin{cases} V_1 : \text{ if } \psi\theta < 0 \\ V_2 : \text{ if } \psi\theta > 0 \end{cases} \tag{7}$$

Figure 2 empirically shows the difference in stability and convergent behavior when using our ζ^{Dirac} for training a Dirac-GAN with a learning rate of $\eta = 0.1$ and Wasserstein loss, versus Sim-GDA or Alt-GDA (the later with many alternate parameter settings n_{d}).

Dirac-GANs with non-infinitesimal learning rates and Wasserstein loss are provably not convergent in general when using Sim-GDA or Alt-GDA Mescheder et al. (2018). Remarkably, ζ^{Dirac} approaches

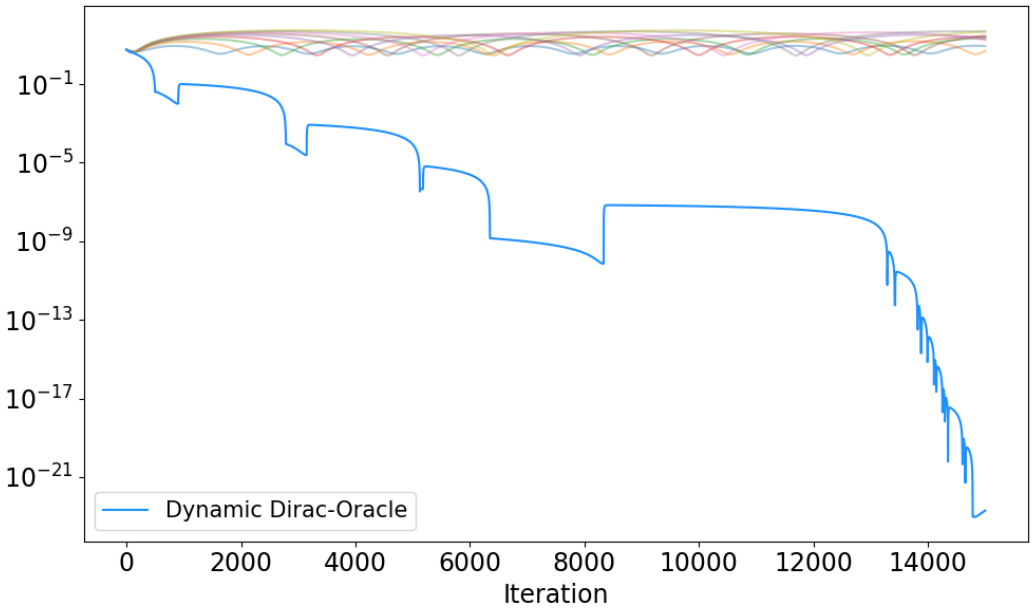

Figure 2: Distance of Dirac-GANs from the saddle point $(0, 0)$ using Wasserstein loss, a loss proven by Mescheder et al. (2018) to be generally not convergent when using Sim-GDA or Alt-GDA. Despite this fact, the Dirac-GAN trained with ζ^{Dirac} (darkblue) approaches the saddle point over 20 orders of magnitude closer than Sim-GDA or any Alt-GDA variation (other lines) tested.

the saddle point to an incredible distance of 10^{-21}. Additionally, ζ^{Dirac} achieves greater stability in its oscillations when close: a phenomenon both Mescheder et al. (2017) and Kodali et al. (2017) showed to be remarkably difficult for non-infinitesimal learning rates.

The success of ζ^{Dirac} lies in the insight to use the relationship between θ and ψ to determine what direction the synthetic distribution would shift if we were to update the generator's parameter. In scenarios in which $\text{sign}(\theta) \neq \text{sign}(\psi)$, updating θ will shift the synthetic closer to the real distribution, whereas when $\text{sign}(\theta) = \text{sign}(\psi)$, updating θ would have the synthetic distribution begin to diverge away. Consequently, ζ^{Dirac} prevents $\mathbb{P}_\theta$ from having divergent behavior.

Theorem 3.1. *Dirac-GANs trained with Wasserstein or BCE loss via ζ^{Dirac} are both Lyapunov stable for both infinitesimal and non-infinitesimal learning rates, and have a tighter bound on stability around saddle-points than Alt-GDA or Sim-GDA for non-infinitesimal learning rates.*

While seemingly no more sophisticated than Sim-GDA or Alt-GDA, the ζ^{Dirac} decision oracle is surprisingly more effective. Where ζ^{Dirac} operates based on training dynamics, Sim-GDA and Alt-GDA instead operate with rigid time-based decisions. Theorem 3.1 shows that ζ^{Dirac} approaches the saddle point stably, and oscillates in a smaller radius around the saddle point than Alt-GDA or Sim-GDA. The proof of Theorem 3.1 can be found in Section D of the supplement. For more details, see Section C in appendix.

4 GENERALIZING OUR DIRAC-GAN SOLUTION

While the capabilities of the dynamic dirac-oracle proposed in Section 3 are impressive in Dirac-GAN settings, we now extend this notion of decision oracles to make them applicable for complex GANs used in practice. GANs generally learn more difficult real distributions, use neural net architectures for the generator and discriminator, and employ mini-batched training. To extend the success of the Dirac-Oracle to practical settings, we show that there are several critical properties of ζ^{Dirac} that can be generalized to construct a successful decision oracle for general GAN optimization. Also, see the supplement for more details.

4.1 EXTENDING TO DEEP NEURAL NET ARCHITECTURES

When dealing with much larger deep neural models for the generator and discriminator, using the simple heuristics like comparing the values of two learned parameters (ie. $\theta < \psi$) becomes a somewhat meaningless and non-interpretable signal. However, in the Dirac-GAN scenario, the relationship between θ and ψ in the parameter-space revealed information about the relationship between the real and synthetic distributions in the feature-space.

By taking this idea of having an interpretable understanding of the relation of distributions in the feature-space via a relationship in the parameter-space, we can show that peering into the *discriminator's projection space* accomplishes this goal for an arbitrary g and d. As convention, discriminators follow the form of being a non-linear 1-dimensional projection of the feature-space: $d : \Omega_x \rightarrow 1$. Also, the goal of GAN loss functions is to encourage a θ such that the synthetic distribution is maximized in the projection of d.

Theorem 4.1. *For a discriminator d, and real/synthetic data $\hat{x} \sim \mathbb{P}_\theta, x \sim \mathbb{P}_\mathbf{r}$, if $\forall \hat{x}, x : d(\hat{x}; \psi) > d(x; \psi)$ then the divergence between $\mathbb{P}_\theta$ and $\mathbb{P}_\mathbf{r}$ will generally not decrease if θ updates according to gradient descent.*

Theorem 4.1 shows that there is a clear link between the relationship real and synthetic distributions have in the discriminator's projection-space and their relationship in the feature-space. Moreover, in the Dirac-GAN case, ζ^{Dirac} chooses only to update ψ when $\psi\theta > 0$. We can see trivially that $\psi\theta > 0 \implies d(\hat{x}; \psi) > d(x; \psi)$, ensuring the assumptions for Theorem 4.1 do *not* hold, and the divergence between $\mathbb{P}_\theta$ and $\mathbb{P}_\mathbf{r}$ decreases when updating θ.

4.2 EXTENDING TO DISTRIBUTIONS WITH NON-ZERO SUPPORT

When dealing with more complicated real distributions than a simple dirac-delta, we need to extend the reasoning of the dynamic dirac-oracle when the real and synthetic distributions have non-zero

supports. As we showed in the previous section Sec 4.1, in situations where the real and synthetic distributions are disjoint, $\mathbb{P}_\theta \cap \mathbb{P}_\mathbf{r} = \emptyset$, then we can confidently update the parameters of the generator θ so long as $\forall \hat{x} \sim \mathbb{P}_\theta, x \sim \mathbb{P}_\mathbf{r} : d(\hat{x}; \psi) < d(x; \psi)$. However, we must additionally consider when $\mathbb{P}_\theta$ and $\mathbb{P}_\mathbf{r}$ are not disjoint.

Lemma 4.2. *If the real and synthetic distributions are not disjoint in the feature-space, then the projections of these distributions in an optimal discriminator space will also be not disjoint:* $\mathbb{P}_\theta \cap \mathbb{P}_\mathbf{r} \neq \emptyset \implies d(\mathbb{P}_\theta; \psi^*) \cap d(\mathbb{P}_\mathbf{r}; \psi^*) \neq \emptyset$.

Lemma 4.3. *If a discriminator is optimal with respect to a fixed generator, and the real and synthetic distributions remain not disjoint in the discriminator's projection-space, then these distributions are not disjoint in the feature-space.*

Lemma 4.2 shows that overlapping distributions in the feature space will remain overlapping in the discriminator's projection-space. However, if we aim to utilize the discriminator's projection-space alone, this will not be sufficient. Lemma 4.3 proves that we can only ensure the distributions overlap in the feature-space so long as the distributions overlap in the discriminator's projection-space and we ensure the discriminator remains locally optimal.

Theorem 4.4. *If a discriminator is optimal with respect to a fixed generator, and that* $\mathbb{P}_\mathbf{f}$ *and* $\mathbb{P}_\mathbf{r}$ *are not disjoint in the discriminator's projection-space, and when* θ *updates according to gradient descent, the divergence between* $\mathbb{P}_\mathbf{f}$ *and* $\mathbb{P}_\mathbf{r}$ *decreases in the discriminator's projection-space, then the divergence between* $\mathbb{P}_\mathbf{f}$ *and* $\mathbb{P}_\mathbf{r}$ *will decrease in the feature-space.*

Using the insights from Lemma 4.3 and Theorem 4.4 to determine under what conditions we can train the generator when the distributions overlap in the discriminator's projection-space, we can ensure $\mathbb{P}_\theta$'s approach towards $\mathbb{P}_\mathbf{r}$ is more stable for complicated distributions of $\mathbb{P}_\mathbf{r}$.

We can see a special case of Theorem 4.4 in the Dirac-GAN, as the Wasserstein divergence in the feature-space is exactly $(\theta - \psi)$. ζ^{Dirac} ensures θ is never updated once $\psi\theta > 0$, as the divergence would increase; namely, $(\theta - \eta\nabla_\theta \mathcal{L}) - \psi > \theta - \psi$ *for any learning rate* $\eta > 0$.

4.3 Extending to Mini-batched Training Schemes

In practice, it is prohibitively expensive to calculate the loss function with respect to the entire dataset in each iteration. Thus, conventional GAN training — and most deep learning — uses mini-batched training Gui et al. (2021).

Mini-batched training comes with a particularly challenging task for generalizing beyond the Dirac-Oracle, as mini-batching removes the ability to observe the entirety of the real and synthetic distributions each iteration. To overcome this, we first notice that when the parameters of the generator

are updated, the projection of the real distribution in the discriminator space remains unchanged. In contrast, when the parameters of the discriminator are updated, the projection of both the real and synthetic distributions may be altered. To leverage this insight in mini-batched training, when taking a batch of real data, we can re-use the projections from the previous batch of real data so long as the previous update only affected θ.

Additionally, when using mini-batches, it can be challenging to efficiently determine when we have reached a locally optimal discriminator for Theorem 4.4. To circumvent this, we inspect the rate at which the divergence between the real and synthetic is growing or shrinking along with the previously chosen update field according to a threshold. By having the discriminator threshold be slightly larger than 1, and the generator threshold be slightly smaller than 1, we can tune these values to have an optimization that balances general stability and aggressiveness to reach equilibria.

5 Our Complete Solution: Dynamic-GDA

In this section, we present our complete solution, Dynamic-GDA, and demonstrate theoretically that Dynamic-GDA boasts strong stability guarantees. Building off the dynamic dirac-oracle described in Section 3, and using the generalizations of each of its components in Section 4, we distill these key ideas to propose Dynamic-GDA, a new method for GAN optimization, as a decision oracle. (Algo. 2).

In every iteration of training, Dynamic-GDA takes in the discriminator projections of a real and synthetic mini-batch. If these two sampled distributions are disjoint in the discriminator-space and the supremum of the synthetic projections is less than the infimum of the real projections, then we update the parameters of the generator (V_1). Likewise, if these two sampled distributions are disjoint in the discriminator-space and the infimum of the synthetic projections is greater than the supremum of the real projections, then we update the discriminator's parameters (V_2).

In scenarios where the two sampled distributions are not disjoint in the discriminator-space, we look at the divergence between their discriminator projections. First, we look to our prior decision last iteration. If the parameters of the discriminator (V_2) were previously updated, we continue this decision until the rate of growth of divergence falls below c_1. However, if the parameters of the generator (V_1) were previously updated, we update the parameters of the generator (V_1) we continue this decision until the divergence between the projections no longer shrinks by a rate of c_2. In practice, we observe $c_1 = 1.2, c_2 = 0.8$ works best.

Algorithm 2 Dynamic-GDA (Dyn-GDA) Decision Oracle

Require: Real batch, $x \sim \mathbb{P}_{\mathbf{r}}$. Synthetic batch, $\hat{x} \sim \mathbb{P}_{\theta}$. Previous divergence, w_{LAST}. Previous choice, V_{LAST}. Discriminator threshold, c_1. Generator Threshold, c_2.

$w \leftarrow \mathcal{W}(d(x;\psi)||d(\hat{x};\psi))$ \\ Wasserstein Divergence
if V_{LAST} is Null **then**
  Return V_2 \\ Only occurs if its the first iteration
else if $\sup\{d(x;\psi)\} < \inf\{d(\hat{x};\psi))\}$ **then**
  Return V_2
else if $\sup\{d(\hat{x};\psi)\} < \inf\{d(x;\psi)\}$ **then**
  Return V_1
else if $V_{\text{LAST}} = V_1$ **then**
  if $w/w_{\text{LAST}} < c_2$ **then**
    Return V_1
  else
    Return V_2
  end if
else if $V_{\text{LAST}} = V_2$ **then**
  if $w/w_{\text{LAST}} > c_1$ **then**
    Return V_2
  else
    Return V_1
  end if
end if

Similar to the no-regret regularization strategy proposed by Kodali et al. (2017), we instead aim to minimize the regret of decision-making based on the impacts the chosen optimization has on samples in the discriminator's projection.

5.1 Lyapunov Stability of Dynamic-GDA

Theorem 5.1. *GANs trained via Dynamic-GDA are Lyapunov stable. Also, in conditions where Sim-GDA and Alt-GDA are locally convergent, Dyn-GDA is too.*

Theorem 5.1 shows that Dynamic-GDA prevents GAN training from completely diverging during training. Conceptually, the key reason lies in Dynamic-GDA taking advantage of an update field for a period of training iterations until it no longer serves to promote beneficial dynamics.

This in turn attempts to implicitly increase the frequency of updating θ to update the synthetic distribution towards the target distribution quickly. Notably, Alt-GDA is unable to do this stably without running into the discriminator problem Goodfellow et al. (2014) identified in the seminal work that proposed GANs, where the discriminator effectively provides detrimental gradient information to the parameters of the generator by having been no longer near a local optimality with respect to the current generator.

Further, Dynamic-GDA doesn't ever update the generator and discriminator at the same iteration (V_3) to prevent the existence of non-real eigenvalues in the Jacobian of V_3. This property can slow

down the rate of convergence and potentially require extremely small learning rates to remain locally convergent Mescheder et al. (2017).

6 EXPERIMENTAL FINDINGS

In this section, we empirically demonstrate that our Dynamic-GDA solution has high impact in practice, i.e., it improves training when using mini-batched training, non-infinitesimal learning rates, and reasonable iteration counts.

To compare the performance of the Dynamic-GDA decision oracle compared to other methods such as Sim-GDA and Alt-GDA, the only aspect varied in all experiments is the *training optimization method* itself. That is, we train all GANs from scratch with the same exact parameter initialization; all generators sample the latent sample according to an identical random seed; and all discriminators receive mini-batches of real data according to the same seed. We experimented with a rich variety of GAN stabilization methods, namely, unsaturated BCE loss Goodfellow et al. (2014), Wasserstein loss Arjovsky et al. (2017), WGAN+GP Arjovsky & Bottou (2017), R2-GP Mescheder et al. (2018), 0-center gradient penalties Roth et al. (2017), spectral regularizations Miyato et al. (2018), instance noise Sønderby et al. (2016), and Piecewise-GP Bhaskara et al. (2022).

We show that when ranking each optimization method's convergence based on *different fitness scores,* such as Wasserstein distance or FID Heusel et al. (2017) in Tables 1 and 2, that across each (GAN, Dataset) pair, Dynamic-GDA is the best optimization method in aggregate. This result is statistically significant with $p < .02$ according to a Friedman Test. See supplement for all experimental details.

6.1 COMPARATIVE EVALUATION ON 2D DATA SETS

Quantifying the precise distance (θ, ψ) from saddlepoints can be challenging even on low-dimensional datasets. Thus, as a precise measure of how well the synthetic distribution matches the real distribution, we utilize the *Wasserstein distance* between the real and synthetic distributions directly in the feature-space for a variety of popular $2D$-datasets:

Gaussian Ring: A $2D$ ring composed of 8 Gaussians each with centers 0.5 away from the origin and a variance of 0.05. **Circle:** A circle with a radius of 0.5 centered at the origin. **Spiral:** A $2D$ projection of a swiss roll Marsland (2011). **Line Segment:** A line spanning $(-0.5, -0.5)$ to $(0.5, 0.5)$.

For each GAN, two feed-forward networks are used for the generator and discriminator, each with 5 hidden layers of 16 neurons, and train with a learning rate of $2 \cdot 10^{-4}$ for 30,000 iterations.

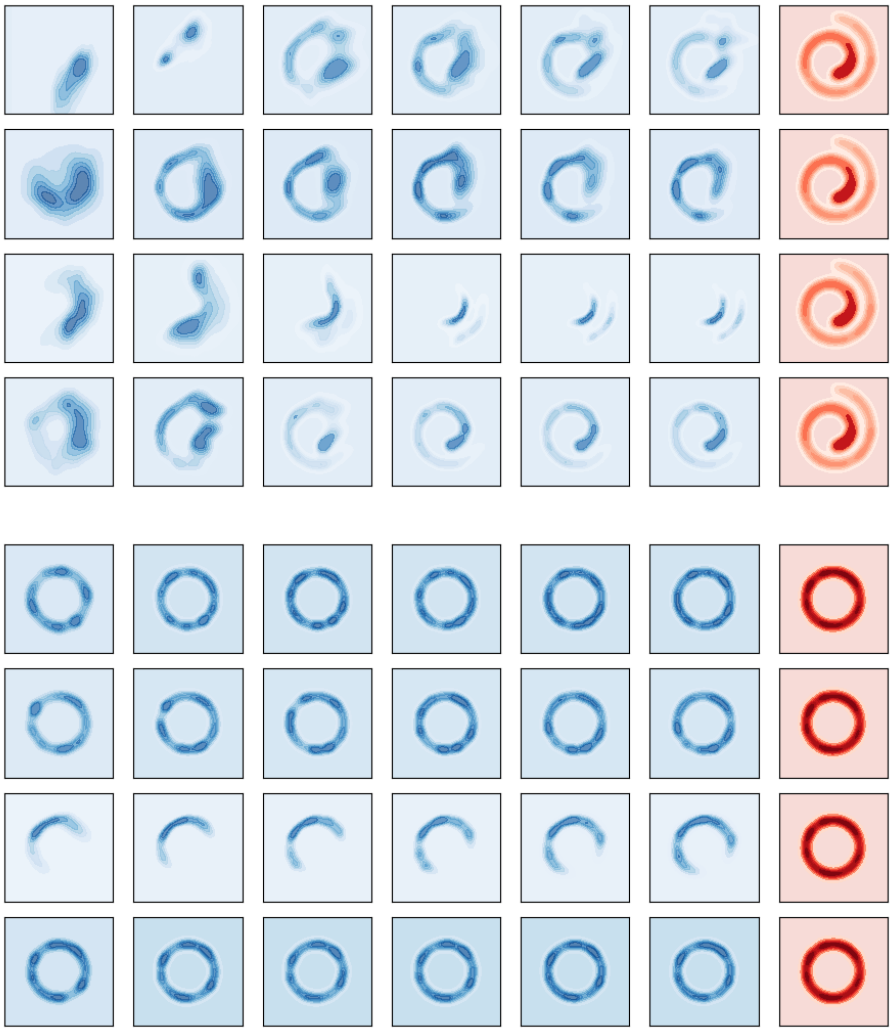

Figure 3: Distributions generated by GANs at (from left to right) 5k, 10k, 15k, 20k, 25k, and 30k iterations when tasked with learning the target distribution (marked red) when equipped with either (from top to bottom) Sim-GDA, Alt-GDA$_{n_d=1}$, Alt-GDA$_{n_d=5}$, or Dyn-GDA. Top block is Spiral, bottom block is Circle.

Table 1 shows the Wasserstein distance at the end of training for GANs trained with each optimization method tested. Dynamic-GDA alone makes up exactly half of all best-ranked optimization methods for the 32 (GAN, $2D$-dataset) pairs. More importantly, the bottom-most row shows the average rank of each optimization method per (GAN, Dataset) pair. Across 3 of the 4 $2D$-datasets, Dynamic-GDA is the most convergent method in aggregate. In the only $2D$-dataset where Dynamic-GDA is not the best ranked in aggregate, it is second best.

Figure 3 shows a visualization of the generated distributions for GANs trained with BCE loss on the Spiral and Circle datasets. See Section G for more visualizations.

6.2 Comparative Evaluation on Image Data Sets

To demonstrate Dynamic-GDA's ability on real-world datasets, we trained a variety of GANs with state-of-the-art stabilization methods on **MNIST** Deng (2012), **CIFAR-10** Krizhevsky et al., and **celebA** Liu et al. (2015) datasets. Figure 5 shows samples of generated images from GANs with each of the optimization methods and BCE loss.

For evaluating the quality of convergence, we use FID score Heusel et al. (2017) to measure the similarity between the real dataset and an equally-sized generated dataset. We use InceptionV3 to calculate FID for CIFAR-10 and celebA, and a pretrained CNN to calculate FID for MNIST.

For the MNIST experiments, we use a feed-forward generator and discriminator architecture each with 5 hidden layers of 1024 neurons. For the CIFAR-10 and celebA experiments, we use an architecture similar to DCGAN Radford et al. (2015) for the generator and discriminator. All GANs were trained for 30,000 iterations with a learning rate of $2 \cdot 10^{-4}$. For instance noise, we use $\mathcal{N}(0, 0.02I)$; for Piecewise-GP $\mathcal{K} = 0.83$, for other regularizations, we use a gradient penalty of 3.

Table 2 shows the FID scores of 8 different GANs stabilization methods at the end of training when equipped with either Sim-GDA, Alt-GDA$_{n_d=1}$, Alt-GDA$_{n_d=5}$, or Dynamic-GDA for each image dataset mentioned. The bottom-most row shows the average rank of each optimization method per (GAN, Dataset) pair. Dynamic-GDA ranks first and second best in aggregate on the MNIST and CIFAR-10 datasets respectively. When considering the rankings across all 56 (GAN, Dataset) pairs, including $2D$ and image together, Dynamic-GDA's rank is statistically significant with $p = .0192$ according to a Friedman Test, further demonstrating Dynamic-GDA's utility for widespread GAN training.

Furthermore, Figure 4 demonstrates's Dyn-GDA strong stability to prevent GANs from undergoing common training pitfalls like mode collapse, unlike Alt-GDA$_{n_d=5}$.

7 Related Work

To our best knowledge, this is the first work to introduce a principled notion of GAN optimization determining the order of the generator and discriminator updates dynamically.

Parameterized Alt-GDA. Goodfellow et al. (2014) first utilized a parameterized Alt-GDA for deciding the specific discriminator-to-generator updating ratio for GAN training. Anecdotal evidence for a 5-1 update ratio is commonly followed in practice Arjovsky et al. (2017).

Architecture-Based Stability. Architecture-specific techniques to improve training stability are for specific domains or data modalities have been explored. DCGAN Radford et al. (2015) has aided in

Table 1: The best optimization method per GAN/dataset pair is **bolded**, second best is underlined. Bottom-most row provides the **average rank** of each optimization method per dataset.

		Wasserstein Distance $\mathcal{W}(\mathbb{P}_\mathbf{r}\|\|\mathbb{P}_\mathbf{f}) \downarrow$			
GAN	Method	Ring	Circle	Spiral	Line
Unsat. BCE (NeurIPS 2014)	Sim-GDA	.3979	.0110	.1089	.0431
	Alt-GDA$_{n_d=1}$	.4740	**.0092**	.2191	.0808
	Alt-GDA$_{n_d=5}$	.0403	.1394	.4959	**.0198**
	Dyn-GDA (ours)	**.0332**	.0120	**.0887**	.0482
Instance Noise (ICLR 2016)	Sim-GDA	.0404	.0125	**.0797**	.1187
	Alt-GDA$_{n_d=1}$	.0504	.0116	.1052	.0692
	Alt-GDA$_{n_d=5}$	**.0266**	.0413	.3338	.0320
	Dyn-GDA (ours)	.0328	**.0075**	.0829	**.0133**
WGAN (ICML 2017)	Sim-GDA	.0341	**.0204**	.0782	.0854
	Alt-GDA$_{n_d=1}$	.0553	.0384	**.0559**	.0417
	Alt-GDA$_{n_d=5}$	**.0433**	.0210	.0811	.0393
	Dyn-GDA (ours)	.0620	.0289	.0750	**.0204**
WGAN+GP (ICLR 2017)	Sim-GDA	.2498	.3566	.4989	.4033
	Alt-GDA$_{n_d=1}$	.2711	.3018	**.3536**	.4719
	Alt-GDA$_{n_d=5}$	.3486	.3156	.6602	.1634
	Dyn-GDA (ours)	**.2276**	**.2702**	.6271	**.0827**
Reg. JS-GAN (NeurIPS 2017)	Sim-GDA	.4024	.0223	.0439	.1219
	Alt-GDA$_{n_d=1}$	.0459	.0143	.0501	.1277
	Alt-GDA$_{n_d=5}$	.0351	**.0124**	.0777	.0204
	Dyn-GDA (ours)	**.0334**	.0167	**.0348**	**.0199**
SN-GAN (ICLR 2017)	Sim-GDA	.0546	.0309	.0908	.1847
	Alt-GDA$_{n_d=1}$	.3377	.0235	.0502	.0383
	Alt-GDA$_{n_d=5}$	.0450	**.0108**	.0693	.0369
	Dyn-GDA (ours)	**.0447**	.0168	**.0418**	**.0212**
R2-GP (ICML 2018)	Sim-GDA	.0730	.0186	.0398	.1359
	Alt-GDA$_{n_d=1}$	.0389	.0177	.0401	.1031
	Alt-GDA$_{n_d=5}$	**.0361**	**.0100**	.0853	**.0137**
	Dyn-GDA (ours)	.0386	.0192	**.0374**	.0229
Piecewise-GP (WACV 2022)	Sim-GDA	.7670	.0146	**.1292**	.1441
	Alt-GDA$_{n_d=1}$	.5262	**.0124**	.1603	.2241
	Alt-GDA$_{n_d=5}$	.0419	.0321	.3191	**.0825**
	Dyn-GDA (ours)	**.0350**	.0126	.1909	.1527
Average Rank	Sim-GDA	3.125	3.125	2.125	3.375
	Alt-GDA$_{n_d=1}$	3.375	**2.250**	2.250	3.500
	Alt-GDA$_{n_d=5}$	1.875	**2.250**	3.750	1.625
	Dyn-GDA (ours)	**1.625**	2.375	**1.875**	**1.500**

image-based GAN training, RNNs in time-series generation Esteban et al. (2017), and GANS Li et al. (2018) for point-cloud generation.

Regularizers for Stability. A variety of regularization techniques has been proposed for improving GAN stability: by adding penalties on discriminator projections of real or synthetic data Roth et al. (2017), adding penalties on the interpolated points between real and synthetic data in the feature-space Arjovsky & Bottou (2017), randomly perturbing real samples Sønderby et al. (2016), or modifying the loss landscape according to the locally observed curvature Mescheder et al. (2017), to name a few. Rigorous analysis of these methods by Mescheder et al. (2018) has shown that certain regularizers can make Sim-GDA and Alt-GDA convergent when initializing (θ_0, ψ_0) near saddle-points.

Support and Mass Alignment. Previous work has proposed techniques to align the supports of the synthetic and real distributions in the feature space Tong et al. (2022). Other works have discussed the impact of gradient saturation affecting convergence Arjovsky & Bottou (2017); Nowozin et al. (2016). They have also demonstrated counterfactual examples to show how divergence minimization

Table 2: The best optmization method per GAN/dataset pair is **bolded**, second best is underlined. Bottom-most row provides the **average rank** of each optimization method per dataset.

GAN	Method	FID Score ↓ celebA	CIFAR-10	MNIST
Unsat. BCE	Sim-GDA	38.477	24.809	142.079
	Alt-GDA$_{n_d=1}$	36.138	**21.550**	149.703
	Alt-GDA$_{n_d=5}$	**32.144**	44.264	1533.967
	Dyn-GDA (ours)	36.911	21.798	**112.754**
Instance Noise	Sim-GDA	**22.000**	28.709	**98.306**
	Alt-GDA$_{n_d=1}$	25.434	**23.407**	175.462
	Alt-GDA$_{n_d=5}$	39.131	47.738	1535.456
	Dyn-GDA (ours)	24.976	24.271	166.755
WGAN	Sim-GDA	65.729	81.072	2058.749
	Alt-GDA$_{n_d=1}$	**31.590**	**39.257**	1839.769
	Alt-GDA$_{n_d=5}$	46.115	53.435	**138.309**
	Dyn-GDA (ours)	34.594	50.276	1828.968
WGAN+GP	Sim-GDA	151.046	135.677	1227.334
	Alt-GDA$_{n_d=1}$	148.783	121.167	**1203.989**
	Alt-GDA$_{n_d=5}$	**145.873**	**103.615**	1968.711
	Dyn-GDA (ours)	379.210	355.324	1979.317
Reg. JS-GAN	Sim-GDA	35.380	23.482	**90.157**
	Alt-GDA$_{n_d=1}$	**33.388**	**23.047**	102.466
	Alt-GDA$_{n_d=5}$	34.637	50.230	1737.525
	Dyn-GDA (ours)	34.427	23.373	95.491
SN GAN	Sim-GDA	54.934	67.036	771.838
	Alt-GDA$_{n_d=1}$	**37.038**	45.970	188.814
	Alt-GDA$_{n_d=5}$	324.535	138.181	222.352
	Dyn-GDA (ours)	50.838	**44.641**	**163.709**
R2-GP	Sim-GDA	38.022	21.591	141.140
	Alt-GDA$_{n_d=1}$	33.671	**21.108**	117.389
	Alt-GDA$_{n_d=5}$	**32.845**	42.108	1882.465
	Dyn-GDA (ours)	34.427	25.421	**111.385**
Piecewise-GP	Sim-GDA	362.825	**27.607**	2477.560
	Alt-GDA$_{n_d=1}$	357.894	33.988	2314.778
	Alt-GDA$_{n_d=5}$	**159.481**	129.076	**2250.221**
	Dyn-GDA (ours)	470.384	29.477	2259.416
Average Rank	Sim-GDA	3.25	2.750	2.625
	Alt-GDA$_{n_d=1}$	**1.750**	**1.500**	2.500
	Alt-GDA$_{n_d=5}$	2.125	3.500	3.000
	Dyn-GDA (ours)	2.875	2.250	**1.875**

in the feature-space alone can lead to spurious optimizations that do not result in finding equilibria Fedus et al. (2017).

8 CONCLUSION

We have demonstrated that altering the order in which the generator and discriminator are updated can have a significant impact on convergence in GAN training. Previous strategies for improving training stability, that had previously been proven non-convergent when built upon Alt-GDA or Sim-GDA, are now proven to be stable when equipped with an adaptive update ordering. Our proposed optimization method Dynamic-GDA not only improves GAN training in isolation but also bolsters additional GAN stabilization strategies when used in tandem. This points at future work into developing better decision oracles with the potential to benefit the body of GAN research and generative modeling as a whole.

9 Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none which we feel must be specifically highlighted here.

References

Guy Ackerson and K Fu. On state estimation in switching environments. *IEEE transactions on automatic control*, 15(1):10–17, 1970.

Martin Arjovsky and Léon Bottou. Towards principled methods for training generative adversarial networks. *arXiv preprint arXiv:1701.04862*, 2017.

Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In *International conference on machine learning*, pp. 214–223, 2017.

Aayush Arora and Shantanu. A review on application of gans in cybersecurity domain. *IETE Technical Review*, pp. 433–441, 2022.

Vineeth S Bhaskara, Tristan Aumentado-Armstrong, Allan D Jepson, and Alex Levinshtein. Gran-gan: Piecewise gradient normalization for generative adversarial networks. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 3821–3830, 2022.

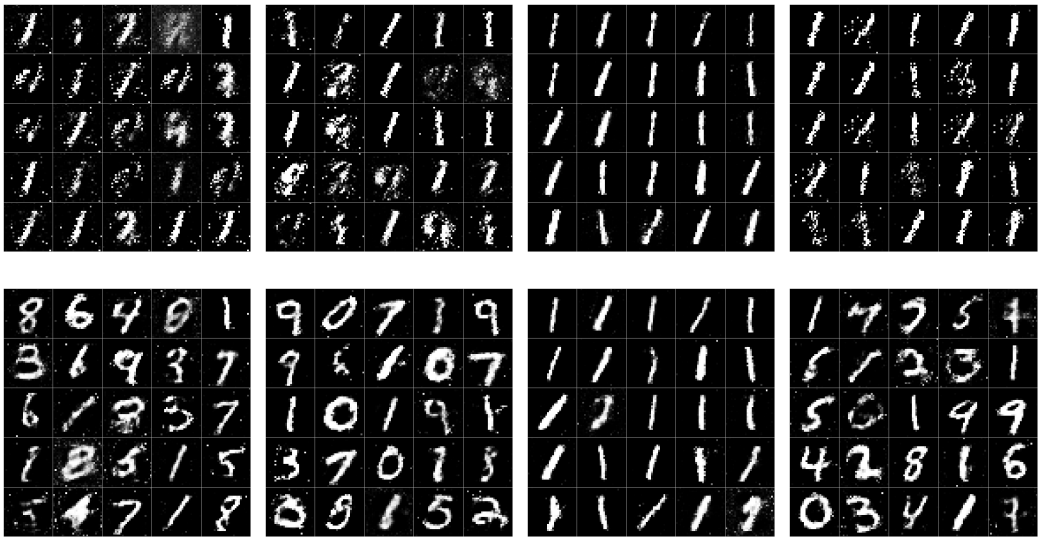

Figure 4: Synthetic samples generated from GANs at 5k (top) and 30k (bottom) iterations when equipped with (from left to right) Sim-GDA, Alt-GDA$_{n_d=1}$, Alt-GDA$_{n_d=5}$, or Dyn-GDA on the MNIST dataset. Where Dyn-GDA and Alt-GDA$_{n_d=5}$ learn the 1's classes earlier than Sim-GDA and Alt-GDA$_{n_d=1}$, Dyn-GDA uniquely does this *without undergoing mode collapse*, and ultimately generates highly diverse samples by the end of training.

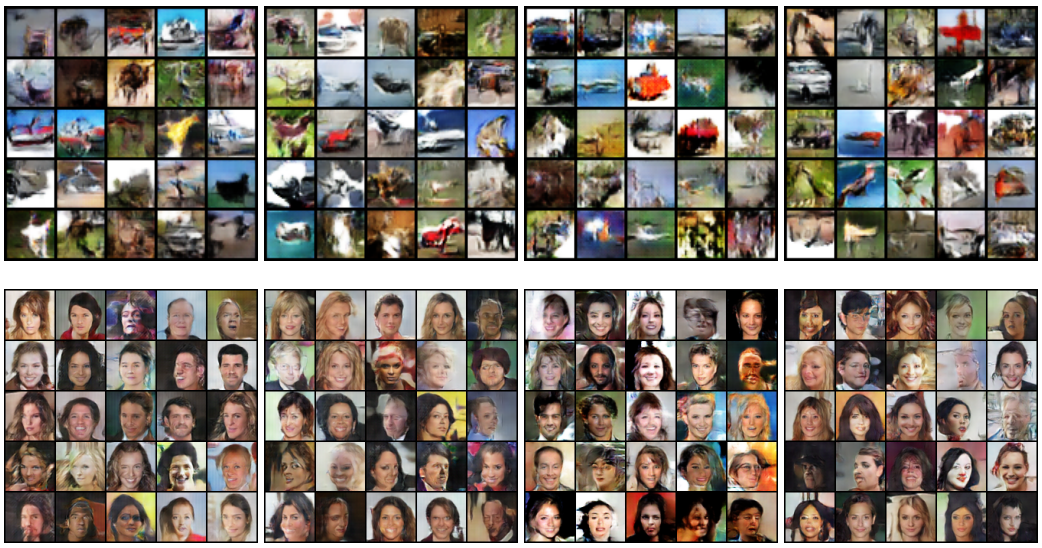

Figure 5: Synthetic samples generated after 30k iterations from GANs equipped with (from left to right) Sim-GDA, Alt-GDA$_{n_d=1}$, Alt-GDA$_{n_d=5}$, or Dyn-GDA during training to learn either the CIFAR-10 (top row) or celebA (bottom row) datasets.

Li Deng. The mnist database of handwritten digit images for machine learning research. *IEEE Signal Processing Magazine*, 29(6):141–142, 2012.

Cristóbal Esteban, Stephanie L Hyland, and Gunnar Rätsch. Real-valued (medical) time series generation with recurrent conditional gans. *arXiv preprint arXiv:1706.02633*, 2017.

William Fedus, Mihaela Rosca, Balaji Lakshminarayanan, Andrew M Dai, Shakir Mohamed, and Ian Goodfellow. Many paths to equilibrium: Gans do not need to decrease a divergence at every step. *arXiv preprint arXiv:1710.08446*, 2017.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 2014.

Jie Gui, Zhenan Sun, Yonggang Wen, Dacheng Tao, and Jieping Ye. A review on generative adversarial networks: Algorithms, theory, and applications. *IEEE transactions on knowledge and data engineering*, pp. 3313–3332, 2021.

Wolfgang Hahn et al. *Stability of motion*, volume 138. Springer, 1967.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017.

Jonathan Ho and Stefano Ermon. Generative adversarial imitation learning. *Advances in neural information processing systems*, 2016.

Naveen Kodali, Jacob Abernethy, James Hays, and Zsolt Kira. On convergence and stability of gans. *arXiv preprint arXiv:1705.07215*, 2017.

Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton. Cifar-10 (canadian institute for advanced research).

Chun-Liang Li, Manzil Zaheer, Yang Zhang, Barnabas Poczos, and Ruslan Salakhutdinov. Point cloud gan. *arXiv preprint arXiv:1810.05795*, 2018.

Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, December 2015.

Stephen Marsland. *Machine learning: an algorithmic perspective*. Chapman and Hall/CRC, 2011.

Lars Mescheder, Sebastian Nowozin, and Andreas Geiger. The numerics of gans. *Advances in neural information processing systems*, 2017.

Lars Mescheder, Andreas Geiger, and Sebastian Nowozin. Which training methods for gans do actually converge? In *International conference on machine learning*, pp. 3481–3490, 2018.

Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. *arXiv preprint arXiv:1802.05957*, 2018.

Vaishnavh Nagarajan and J. Zico Kolter. Gradient descent gan optimization is locally stable. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

Sebastian Nowozin, Botond Cseke, and Ryota Tomioka. f-gan: Training generative neural samplers using variational divergence minimization. *Advances in neural information processing systems*, 2016.

Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, Adam Lerer, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in Neural Information Processing Systems*, 32: 8024–8035, 2019.

Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.

Kevin Roth, Aurelien Lucchi, Sebastian Nowozin, and Thomas Hofmann. Stabilizing training of generative adversarial networks through regularization. *Advances in neural information processing systems*, 2017.

Divya Saxena and Jiannong Cao. Generative adversarial networks (gans) challenges, solutions, and future directions. *ACM Computing Surveys (CSUR)*, pp. 1–42, 2021.

Casper Kaae Sønderby, Jose Caballero, Lucas Theis, Wenzhe Shi, and Ferenc Huszár. Amortised map inference for image super-resolution. *arXiv preprint arXiv:1610.04490*, 2016.

Shangyuan Tong, Timur Garipov, Yang Zhang, Shiyu Chang, and Tommi S Jaakkola. Adversarial support alignment. *arXiv preprint arXiv:2203.08908*, 2022.

Zhengli Zhao, Zizhao Zhang, Ting Chen, Sameer Singh, and Han Zhang. Image augmentations for gan training. *arXiv preprint arXiv:2006.02595*, 2020.

A Extension of Preliminaries

In this section, we will describe previous theoretical contributions by Mescheder et al. (2018) to prove under what assumption can GANs exhibit convergence or at the least stability under a discretized dynamical system.

A.1 Convergence Theory from Prior Work

As described by Nagarajan & Kolter (2017), the vector field of GAN training at any point (θ, ψ) can be described by the gradient vector field $v(\theta, \psi)$:

$$v(\theta, \psi) = \begin{pmatrix} -\nabla_\theta \mathcal{L}(\theta, \psi) \\ \nabla_\psi \mathcal{L}(\theta, \psi) \end{pmatrix} \tag{8}$$

Given that an update operator F follows the form

$$F(\theta, \psi) = I + \eta V(\theta, \psi)$$

for some arbitrary field V, Mescheder et al. (2017) showed that if the Jacobian of an update operator F has eigenvalues with absolute values greater than 1 at the saddle-point, then GAN training will generally not converge. Additionally, if the Jacobian of an update operator F has eigenvalues with absolute values less than 1 at the saddle-point, then GAN training will converge linearly with a rate of $\mathcal{O}(|\lambda_{\text{MAX}}|^k)$, where λ_{MAX} is the eigenvalue with the greatest magnitude from $F^{'}$. Similarly, when all eigenvalues lie on the unit circle, convergence is at best sub-linear.

Mescheder et al. (2017) notably also showed that it is a necessity, but not necessarily sufficient, that the Jacobian of $v(\theta^*, \psi^*)$ has eigenvalues all with a negative real part for there to be linear convergence when training with either Alt-GDA or Sim-GDA.

Mescheder et al. (2018) then showed for GANs trained with a non-infinitesimal learning rate η via Sim-GDA, that training will converge at best sub-linearly if and only if the Jacobian of the update operator

$$F_3(\theta, \psi) = \begin{pmatrix} \theta - \eta \nabla_\theta \mathcal{L}(\theta, \psi) \\ \psi + \eta \nabla_\psi \mathcal{L}(\theta, \psi) \end{pmatrix}$$

has eigenvalues that all have a negative real part at the saddle-point (θ^*, ψ^*), and η must be within the bound seen in Eq. 9 for all eigenvalues λ.

$$\eta_{\text{SIM}} < \frac{1}{|\text{Re}(\lambda)|} \frac{2}{1 + \left(\frac{\text{Im}(\lambda)}{\text{Re}(\lambda)}\right)^2} \tag{9}$$

Similarly, Mescheder et al. (2018) also showed for GANs trained according to Alt-GDA, that $v(\theta^*, \psi^*)$ must have eigenvalues all with a negative real part, and η must be infinitesimally small to ensure the eigenvalues of the Jacobian of the Alt-GDA update operator lie on the unit circle. In terms of update operators F_1 and F_2, the update operator of Alt-GDA is $F_2 \circ F_1$, where

$$F_1(\theta, \psi) = \begin{pmatrix} \theta - \eta \nabla_\theta \mathcal{L}(\theta, \psi) \\ \psi \end{pmatrix}$$

$$F_2(\theta, \psi) = \begin{pmatrix} \theta \\ \psi + \eta \nabla_\psi \mathcal{L}(\theta, \psi) \end{pmatrix}$$

B LYAPUNOV STABILITY

Lyapunov stability of a dynamic system is a form of stability that ensures a notion of *hovering* around equilibria. While not as strong as asymptotic stability, where every initial condition approaches an equilibrium point as $t \to \infty$, systems with Lyapunov stability prevent initial conditions from diverging infinitely far away from equilibria.

For continuous-time systems, where f is a dynamic system such that $f(\mathbf{x}(t)) = \dot{\mathbf{x}}$, and f has an equilibrium $\mathbf{x}^*$ such that $f(\mathbf{x}^*) = 0$, f is said to be Lyapunov stable, if, for every $\epsilon > 0$, there exists a $\delta > 0$ such that if $||\mathbf{x}(0) - \epsilon|| < \delta$, then for every $t > 0$, $||\mathbf{x}(t) - x^*|| < \delta$. To prove a continuous-time system is Lyapunov stable, a Lyapunov potential function P is used. P can be any potential function such that P is symmetric and positive definite. For a linear system that is defined via the matrix A where $\dot{x} = Ax$, then A is Lyapunov stable if $A^T P + PA$ is negative semi-definite Hahn et al. (1967) for all $x \neq 0$. For nonlinear systems, one must show that there exists a P such that $\nabla P(x) \cdot f(x)$ is negative semi-definite. Continuous-time systems are effective for modeling when training under an infinitesimally small learning rate. For non-infinitesimal learning rates, we use discrete-time systems where a similar definition holds, where a linear system that is defined via the matrix A where $x(k+1) = Ax(k)$, then A is Lyapunov stable if $A^T PA - P$ is negative semi-definite for all $x \neq 0$. For nonlinear systems, one must show that there exists a P such that $P(f(x(k+1))) - P(x(k)$ is negative or non-increasing for all k.

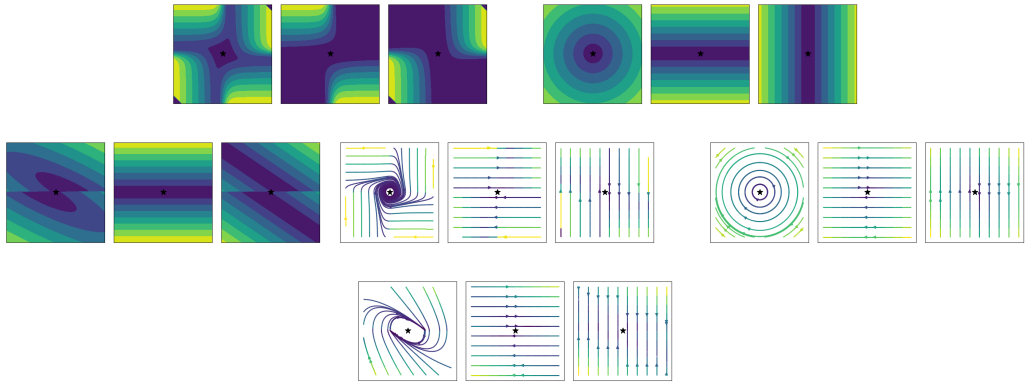

Figure 6: Gradient fields (bottom row) and their norms (top row) of a Dirac GAN equipped with (from left group to right group) a Vanilla, Wasserstein, and Wasserstein with gradient penalty loss, respectively. x-axis is θ and y-axis is ψ. Brighter colors indicate a larger gradient magnitude. The joint gradient space (left) illustrates the instability that can arise in the simplest of GAN formulations, and optimizing both machines in each iteration subjects the GAN's parameters to these undesired unstable dynamics. However, a decomposition of the gradient field into its orthogonal components w.r.t the generator (middle) and discriminator (right) can yield additional control over the joint gradient field. Optimization regimes that utilize their own sequence of choosing one of these 3 fields each iteration through the course of training can seek saddle points more efficiently.

C Utilizing Non-Static Update Functions

One of the critical assumptions of these insights by Mescheder et al. (2017) and Mescheder et a. (2018) is that the chosen update function F must be chosen unilaterally at all initial conditions of (θ_0, ψ_0). Even more so, in the non-infinitesimal case were a learning rate $\eta > 0$ is utilized equipped with the update operator F, η must be sufficiently small for the eigenvalues of F'_η to be in the unit ball for all iterates (θ_k, ψ_k).

An overlooked aspect of the optimization strategy is the use of constraining the min-max optimization to a fixed, premeditated update operator F acting on a fixed field $V(\theta, \psi)$. We then show that when we allow the selection of V to itself to be a function of an oracle ζ conditioned on (θ, ψ), then when a non-infinitesimal learning rate is imposed, $\zeta(\theta, \psi) \to V$, the resulting update operator $F(\theta, \psi)$ can ensure the absolute value of the eigenvalues lie in the unit circle. Figure 6 shows a decomposition of the gradient field of v, and adaptively choosing which update field to use gives greater flexibility in traversing the parameter-space of $\boldsymbol{\theta} \times \boldsymbol{\psi}$.

D Proofs and Remarks

ζ^{Dirac} is an instantiation of a switching dynamic system Ackerson & Fu (1970), where ζ^{Dirac} is equipped with a set of systems $\mathbf{V} = \{V_1, V_2, V_3\}$, and has a conditional update operator, and consequently optimizes along potentially different gradient vector fields each iteration. When we can

define the dynamic system formed by ζ^{Dirac} under an infinitesimal and non-infinitesimal learning rate, each found in Equations 10 and 11 respectively.

$$\zeta^{\text{Dirac}}(\mathbf{x}) = \begin{cases} V_1 \text{ if } \psi\theta < 0 \\ V_2 \text{ if } \psi\theta > 0 \end{cases} \tag{10}$$

$$\zeta^{\text{Dirac}}(\mathbf{x}; \eta) = \begin{cases} \mathbf{x} - \eta V_1 \text{ if } \psi\theta < 0 \\ \mathbf{x} + \eta V_2 \text{ if } \psi\theta > 0 \end{cases} \tag{11}$$

For solving for saddle-points, we impose that we want to solve

$$\nabla_\theta^2 \mathcal{L}(\theta, \psi) < 0 < \nabla_\psi^2 \mathcal{L}(\theta, \psi),$$

$$\nabla_\theta \mathcal{L}(\theta, \psi) = \nabla_\psi \mathcal{L}(\theta, \psi) = 0.$$

So, the equilibrium point lies at $v(0) = 0$ using the gradient vector field in Equation 8.

For a variety of our proofs, we will use the Potential function H in Equation 12,

$$H(\mathbf{x}) = \begin{pmatrix} \frac{1}{2}\theta^2 & 0 \\ 0 & \frac{1}{2}\psi^2 \end{pmatrix} \tag{12}$$

Lemma D.1. *Dirac-GANs trained via ζ^{Dirac} are both Lyapunov stable for infinitesimal learning rates when*

$0 \geq -\theta\nabla_\theta \mathcal{L}(\theta, \psi)$ *so long as* $\psi\theta < 0$*,*

or when $0 \geq \psi\nabla_\psi \mathcal{L}(\theta, \psi)$ *so long as* $\psi\theta > 0$*.*

Proof.

We utilize a Lyapunov potential function H in Equation 12.

Then

$$\nabla H \cdot \zeta^{Dirac} = \begin{cases} \begin{pmatrix} -\theta\mathcal{L}(\theta,\psi) & 0 \\ 0 & 0 \end{pmatrix} : \psi\theta < 0 \\ \\ \begin{pmatrix} 0 & 0 \\ 0 & \psi\mathcal{L}(\theta,\psi) \end{pmatrix} : \psi\theta > 0 \end{cases}$$

Thus, we can see that for a given $\mathcal{L}$, $\nabla H(\theta,\psi) \cdot \zeta^{Dirac}(\theta,\psi)$ *is negative semi definite so long as*

$0 \geq -\theta\mathcal{L}(\theta,\psi)$ *when* $\psi\theta < 0$

and

$0 \geq \psi\mathcal{L}(\theta,\psi)$ *when* $\psi\theta > 0$

□

Lemma D.2. *Dirac-GANs trained via* ζ^{Dirac} *are both Lyapunov stable for non-infinitesimal learning rates*

$$\eta \leq \frac{\theta^2 - 2\theta}{2\nabla_\theta \mathcal{L}(\theta,\psi)} \text{ when } 0 \geq \frac{-\psi^2}{2} \text{ and } \psi\theta < 0$$

$$\eta \leq \frac{\psi^2 - 2\psi}{2\nabla_\psi \mathcal{L}(\theta,\psi)} \text{ and } 0 \geq \frac{-\theta^2}{2} \text{ when } \psi\theta > 0$$

Proof.

We utilize a Lyapunov potential function H *in Equation 12.*

Then to follow the form $P(f(x(k+1))) - P(x(k) =$, *then* $H(\zeta^{Dirac}(\theta_{k+1},\psi_{k+1})) - H(\theta,\psi) =$

$$\begin{cases} \begin{pmatrix} \theta - \eta\nabla_\theta\mathcal{L}(\theta,\psi) - \frac{\theta^2}{2} & 0 \\ 0 & \frac{-\psi^2}{2} \end{pmatrix} : \psi\theta < 0 \\ \\ \begin{pmatrix} \frac{-\theta^2}{2} & 0 \\ 0 & \psi + \eta\nabla_\psi\mathcal{L}(\theta,\psi) - \frac{\psi^2}{2} \end{pmatrix} : \psi\theta > 0 \end{cases}$$

Then we see

$0 \geq \theta - \eta\nabla_\theta\mathcal{L}(\theta,\psi) - \frac{\theta^2}{2}$ *and* $0 \geq \frac{-\psi^2}{2}$ *when* $\psi\theta < 0$

$0 \geq \psi + \eta \nabla_\psi \mathcal{L}(\theta, \psi) - \frac{\psi^2}{2}$ *and* $0 \geq \frac{-\theta^2}{2}$ *when* $\psi\theta > 0$

We can rewrite this in terms of η*:*

$$\psi\theta < 0 \implies \eta \leq \frac{\theta^2 - 2\theta}{2\nabla_\theta \mathcal{L}(\theta, \psi)} \text{ and } 0 \geq \frac{-\psi^2}{2}$$

$$\psi\theta > 0 \implies \eta \leq \frac{\psi^2 - 2\psi}{2\nabla_\psi \mathcal{L}(\theta, \psi)} \text{ and } 0 \geq \frac{-\theta^2}{2}$$

□

Lemma D.3. *Dirac-GANs trained with Wasserstein loss via* ζ^{Dirac} *are Lyapunov stable for an infinitesimal learning rate.*

Proof.

Wasserstein loss is defined by $\mathcal{L}(\theta, \psi) = -\psi\theta$ *and it follows that*

$\nabla_\theta \mathcal{L}(\theta, \psi) = -\psi$

$\nabla_\psi \mathcal{L}(\theta, \psi) = -\theta$

Thus, when considering that for this $\mathcal{L}$*,* $\zeta^{Dirac}_{\mathbf{V}}$ *requires*

$\psi\theta < 0 \implies 0 \geq -\theta \nabla_\theta \mathcal{L}(\theta, \psi)$

$\psi\theta > 0 \implies 0 \geq \psi \nabla_\psi \mathcal{L}(\theta, \psi)$*,*

To maintain Lyapunov stability. We see that

$\psi\theta < 0 \implies 0 \geq \theta\psi$

$\psi\theta > 0 \implies 0 \geq -\psi\theta$*,*

Holds trivially. Since Dirac-GANs with ζ^{Dirac} *are Lyapunov stable according to Lemma D.1, we have demonstrated* ζ^{Dirac} *will coerce Dirac-GANs trained with Wasserstein loss to be Lyapunov stable.*
□

Lemma D.4. *Dirac-GANs trained with BCE loss via* ζ^{Dirac} *are Lyapunov stable for an infinitesimal learning rate.*

Proof.

BCE loss is defined for Dirac-GANs

$$\mathcal{L}(\theta, \psi) = \log(\sigma(\psi \cdot 0)) + \log(1 - \sigma(\psi\theta))$$

where σ is the sigmoid activation function with property $\sigma^{'}(x) = \sigma(x)(1-\sigma(x))x^{'}$

and it follows that

$$\nabla_\theta \mathcal{L}(\theta, \psi) = \frac{\sigma(\psi\theta)(1-\sigma(\psi\theta))\psi}{1-\sigma(\psi\theta)}$$

$$\nabla_\psi \mathcal{L}(\theta, \psi) = \frac{\sigma(\psi\theta)(1-\sigma(\psi\theta))\theta}{1-\sigma(\psi\theta)}$$

Thus, when considering that for this $\mathcal{L}$, $\zeta_{\mathbf{V}}^{Dirac}$ requires

$\psi\theta < 0 \implies 0 \geq -\theta\nabla_\theta \mathcal{L}(\theta, \psi)$

$\psi\theta > 0 \implies 0 \geq \psi\nabla_\psi \mathcal{L}(\theta, \psi)$,

To maintain Lyapunov stability per Lemma D.1. We see that

$\psi\theta < 0 \implies 0 \geq -\theta\psi\sigma(\psi\theta)$

$\psi\theta > 0 \implies 0 \geq \psi\theta\sigma(\psi\theta)$,

Holds trivially since $\sigma(\psi\theta)$ is striclty bounded between the exclusive interval $(0, 1)$. Since Dirac-GANs with ζ^{Dirac} are Lyapunov stable according to Lemma D.1, we have demonstrated ζ^{Dirac} will coerce Dirac-GANs trained with BCE loss to be Lyapunov stable. □

Lemma D.5. *Dirac-GANs trained with Wasserstein loss via ζ^{Dirac} are Lyapunov stable for a non-infinitesimal learning rate η where*

$$\eta \leq 1 - \frac{\theta}{2} \text{ when } \zeta^{Dirac} \to V_1$$

$$\eta \leq 1 - \frac{\psi}{2} \text{ when } \zeta^{Dirac} \to V_2$$

Proof.

Wasserstein loss is defined by $\mathcal{L}(\theta, \psi) = -\psi\theta$ and it follows that

$$\nabla_\theta \mathcal{L}(\theta, \psi) = -\psi$$

$$\nabla_\psi \mathcal{L}(\theta, \psi) = -\theta$$

Thus, when considering that for this $\mathcal{L}$, $\zeta_{\mathbf{V}}^{Dirac}$ requires η to satisfy the inequalities:

$$\psi\theta < 0 \implies \eta \leq \frac{\theta^2 - 2\theta}{2\nabla_\theta \mathcal{L}(\theta, \psi)} \text{ and } 0 \geq \frac{-\psi^2}{2}$$

$$\psi\theta > 0 \implies \eta \leq \frac{\psi^2 - 2\psi}{2\nabla_\psi \mathcal{L}(\theta, \psi)} \text{ and } 0 \geq \frac{-\theta^2}{2}$$

to maintain Lyapunov stability per Lemma D.2. We see that

$$\psi\theta < 0 \implies \eta \leq 1 - \frac{\theta}{2}, 0 \geq \frac{-\psi^2}{2}$$

$$\psi\theta > 0 \implies \eta \leq 1 - \frac{\psi}{2}, 0 \geq \frac{-\theta^2}{2}$$

Holds for relatively large η even when $||(\theta, \psi) - (0, 0)||$ is small. Since Dirac-GANs with ζ^{Dirac} are Lyapunov stable according to Lemma D.2, we have demonstrated ζ^{Dirac} will coerce Dirac-GANs trained with Wasserstein loss to be Lyapunov stable for sizeable, non-infinitesimal learning rates. □

Lemma D.6. *Dirac-GANs trained with BCE loss via ζ^{Dirac} are Lyapunov stable for a non-infinitesimal learning rate η where*

Proof.

BCE loss is defined for Dirac-GANs

$$\mathcal{L}(\theta, \psi) = \log(\sigma(\psi \cdot 0)) + \log(1 - \sigma(\psi\theta))$$

where σ is the sigmoid activation function with property $\sigma^{'}(x) = \sigma(x)(1 - \sigma(x))x^{'}$

and it follows that

$$\nabla_\theta \mathcal{L}(\theta, \psi) = \frac{\sigma(\psi\theta)(1 - \sigma(\psi\theta))\psi}{1 - \sigma(\psi\theta)}$$

$$\nabla_\psi \mathcal{L}(\theta, \psi) = \frac{\sigma(\psi\theta)(1 - \sigma(\psi\theta))\theta}{1 - \sigma(\psi\theta)}$$

Thus, when considering that for this $\mathcal{L}$, $\zeta_{\mathbf{V}}^{Dirac}$ requires η to satisfy the inequalities:

$$\psi\theta < 0 \implies \eta \leq \frac{\theta^2 - 2\theta}{2\nabla_\theta \mathcal{L}(\theta, \psi)} \text{ and } 0 \geq \frac{-\psi^2}{2}$$

$$\psi\theta > 0 \implies \eta \leq \frac{\psi^2 - 2\psi}{2\nabla_\psi \mathcal{L}(\theta, \psi)} \text{ and } 0 \geq \frac{-\theta^2}{2}$$

to maintain Lyapunov stability per Lemma D.2. We see that

$$\psi\theta < 0 \implies \eta \leq \frac{\theta^2 - 2\theta}{2\sigma(\psi\theta)\psi} \text{ and } 0 \geq \frac{-\psi^2}{2}$$

$$\psi\theta > 0 \implies \eta \leq \frac{\psi^2 - 2\psi}{2\sigma(\psi\theta)\theta} \text{ and } 0 \geq \frac{-\theta^2}{2}$$

Holds for a relatively large η even when $||(\theta, \psi) - (0, 0)||$ is large, as the bound on $\sigma(\psi\theta)$ will saturate the denominator when $\psi\theta < 0$. Since Dirac-GANs with ζ^{Dirac} are Lyapunov stable according to Lemma D.2, we have demonstrated ζ^{Dirac} will coerce Dirac-GANs trained with BCE loss to be Lyapunov stable. □

Theorem 3.1 Dirac-GANs trained with Wasserstein or BCE loss via ζ^{Dirac} are both Lyapunov stable for both infinitesimal and non-infinitesimal learning rates, and have a tighter bound on stability around saddle-points than Alt-GDA or Sim-GDA for non-infinitesimal learning rates.

Proof.

We prove ζ^{Dirac} is Lyapunov stable for Wasserstein GANs with infinitesimal learning rates in Lemma D.3, and non-infinitesimal learning rates in Lemma D.5.

Similarly, we prove ζ^{Dirac} is Lyapunov stable for BCE GANs with infinitesimal learning rates in Lemma D.4, and non-infinitesimal learning rates in Lemma D.6.

Mescheder et al. (2018) proved that Sim-GDA is not stable near equilibria for Dirac-GANs even when an infinitesimal learning rate is used. Worse, Sim-GDA diverge will all non-infinitesimal learning rates.

Mescheder et al. (2018) also shows that for Dirac-GANs equipped with Alt-GDA can only achieve non-divergent behavior when

$$\eta \leq \frac{2}{\sqrt{n_g n_d} \nabla \mathcal{L}(\theta, \psi)} \tag{13}$$

where n_g is the number of generator updates after performing n_d discriminator updates. The inequality of η in Equation 13 places tighter bounds on sufficient $\eta's$ than the bound place by ζ^{Dirac}:

$$\eta \leq \frac{\theta^2 - 2\theta}{2\nabla_\theta \mathcal{L}(\theta, \psi)}$$

when updating θ, and

$$\eta \leq \frac{\psi^2 - 2\psi}{2\nabla_\psi \mathcal{L}(\theta, \psi)}$$

when updating ψ. $\square$

Theorem 4.1. *For a discriminator d, and real/synthetic data $\hat{x} \sim \mathbb{P}_\theta, x \sim \mathbb{P}_\mathbf{r}$, if $\forall \hat{x}, x : d(\hat{x}; \psi) > d(x; \psi)$ then the divergence between $\mathbb{P}_\theta$ and $\mathbb{P}_\mathbf{r}$ will generally not decrease if θ updates according to gradient descent.*

Proof.

If a given d is a Wasserstein critic, then, the locally optimal critic will asymptotically send $d(\hat{x}; \psi)$ towards $-\infty$ and send $d(x; \psi)$ towards ∞ per the definition of Wasserstein loss. Thus, in the asymptotics of optimizing ψ, in the hopes of ultimately approaching a locally optimal discriminator, we expect

$$\mathbb{E}_{\hat{x} \sim \mathbb{P}_\theta} d(\hat{x}; \psi) - \mathbb{E}_{x \sim \mathbb{P}_\mathbf{r}} d(x; \psi)$$

to increase. Thus, if we were then to optimize θ, we would expect to observe θ iteratively maximize the projection of $\mathbb{P}_\theta$. However, if $\mathbb{E}_{\hat{x} \sim \mathbb{P}_\theta} d(\hat{x}; \psi) > \mathbb{E}_{\hat{x} \sim \mathbb{P}_\theta} d(\hat{x}; \psi)$ then there are erroneous regions in the feature-space d projects to a greater degree outside the support of $\mathbb{P}_\mathbf{r}$. Therefore, if in extreme cases, $\inf\{d(\hat{x}; \psi)\} > \sup\{d(x; \psi)\}$, then θ will be optimized such that it travels towards erroneously project regions. This has no guarantee on improving the divergence between $\mathbb{P}_\theta$ and $\mathbb{P}_\mathbf{r}$, and may even cause detrimental, divergent behavior of $\mathbb{P}_\theta$.

Furthermore, for bounded discriminators such as the discriminators employed for use with BCE loss, it is well known that a locally optimal discriminator's projection converges towards the real-to-synthetic mass ratio at each point in the feature space. Thus, when $\mathbb{E}_{\hat{x} \sim \mathbb{P}_\theta} d(\hat{x}; \psi) > \mathbb{E}_{x \sim \mathbb{P}_\mathbf{r}} d(x; \psi)$ or in the extreme case, $\inf\{d(\hat{x}; \psi)\} > \sup\{d(x; \psi)\}$, then the discriminator's projection of the real-to-synthetic mass ration cannot be faithful to the feature-space, and may cause $\mathbb{P}_\theta$ to converge to towards erroneous regions with zero density of $\mathbb{P}_\mathbf{r}$. $\square$

Lemma 4.2 *If the real and synthetic distributions are not disjoint in the feature-space, then the projections of these distributions in the discriminator space will also be not disjoint:* $\mathbb{P}_\theta \cap \mathbb{P}_\mathbf{r} \neq \emptyset \implies d(\mathbb{P}_\theta; \psi) \cap d(\mathbb{P}_\mathbf{r}; \psi) \neq \emptyset$.

Proof.

We prove this by contraction.

Let's first assume the contradictory statement, where $\mathbb{P}_\theta$ *and* $\mathbb{P}_\mathbf{r}$ *are non-disjoint in the feature-space, but* $d(\mathbb{P}_\theta; \psi)$ *and* $d(\mathbb{P}_\mathbf{r}; \psi)$ *are disjoint. This implies there is a point* p *that lies in the support of* $\mathbb{P}_\mathbf{r}$ *and the support of* $\mathbb{P}_\theta$ *simultaneously.*

Since $d(\mathbb{P}_\theta; \psi)$ *and* $d(\mathbb{P}_\mathbf{r}; \psi)$ *are disjoint, there does not exist a point* q*, where* q *lies in the support of* $d(\mathbb{P}_\theta; \psi)$ *and* $d(\mathbb{P}_\mathbf{r}; \psi)$ *simultaneously.*

However, since d *is a continuous operator on the feature-space, infinitesimal perturbations in the feature-space will be infinitesimally small in the image of* d*.*

This is a contradiction since the lack of existence of q *violates the continuity assumption of* d*. Therefore,* $d(\mathbb{P}_\theta; \psi)$ *and* $d(\mathbb{P}_\mathbf{r}; \psi)$ *must not be disjoint.* □

Lemma 4.3 *If a discriminator is optimal concerning a fixed generator, and the real and synthetic distributions remain not disjoint in the discriminator's projection-space, then these distributions are not disjoint in the feature-space.*

Proof.

We will prove this by contradiction.

Let's first assume the contradictory statement, where $\mathbb{P}_\theta$ *and* $\mathbb{P}_\mathbf{r}$ *are non-disjoint in the feature-space.*

When we optimize ψ*, the extremum that* ψ *reaches the projection of* d *such that* $d(\mathbb{P}_\theta; \psi)$ *and* $d(\mathbb{P}_\mathbf{r}; \psi)$ *is maximally separated. If* $d(\mathbb{P}_\theta; \psi)$ *and* $d(\mathbb{P}_\mathbf{r}; \psi)$ *remain not disjoint at the locally extremum for a fixed* θ *then there must exist a point* p *such that* p *lies in the support of* $d(\mathbb{P}_\mathbf{r}; \psi)$ *and the support of* $d(\mathbb{P}_\theta; \psi)$ *simultaneously.*

Since d *is a continuous operator on the feature-space that has aimed to maximize the distance between* $d(\mathbb{P}_\mathbf{r}; \psi)$ *and* $d(\mathbb{P}_\theta; \psi)$*, any infinitesimal perturbations in the feature-space will be infinitesimally small in the image of* d*.*

This is a contradiction since there must exist a point q *that exists in the supports of* $\mathbb{P}_\theta$ *and* $\mathbb{P}_\mathbf{r}$ *simultaneously. By assuming* $\mathbb{P}_\theta$ *and* $\mathbb{P}_\mathbf{r}$ *were disjoint, we have violated the continuity assumption of* d*. Therefore,* $\mathbb{P}_\theta$ *and* $\mathbb{P}_\mathbf{r}$ *must not be disjoint.* □

Theorem 4.4. *If a discriminator is optimal concerning a fixed generator, and that $\mathbb{P}_{\mathbf{f}}$ and $\mathbb{P}_{\mathbf{r}}$ are not disjoint in the discriminator's projection-space, and when θ updates according to gradient descent the divergence between $\mathbb{P}_{\mathbf{f}}$ and $\mathbb{P}_{\mathbf{r}}$ decreases in the discriminator's projection-space, then the divergence between $\mathbb{P}_{\mathbf{f}}$ and $\mathbb{P}_{\mathbf{r}}$ will decrease in the feature-space.*

Proof.

We will prove this by contradiction.

First consider the local optimality of the discriminator for a fixed generator. Once locally optimal, the discriminator has achieved maximum separation between $d(\mathbb{P}_{\mathbf{r}}; \psi^*)$ and $d(\mathbb{P}_\theta; \psi^*)$ in the discriminator's projection. If $d(\mathbb{P}_{\mathbf{r}}; \psi^*)$ and $d(\mathbb{P}_\theta; \psi^*)$ are then disjoint, then we can point to Lemma 4.3 to show that the distributions must then be disjoint. Here, we instead consider when $d(\mathbb{P}_{\mathbf{r}}; \psi^*)$ and $d(\mathbb{P}_\theta; \psi^*)$ are still not disjoint. Then there exists a smooth path due to the continuity of d for θ to update.

If we observe that for a single update iteration f θ according to an infinitesimally smaller learning rate decreased the divergence between $d(\mathbb{P}_{\mathbf{r}}; \psi^*)$ and $d(\mathbb{P}_\theta; \psi^*)$, but increase the divergence between $\mathbb{P}_{\mathbf{f}}$ and $\mathbb{P}_{\mathbf{r}}$ in the feature space, then we should observe either less synthetic mass in the support of $\mathbb{P}_{\mathbf{r}}$, or redistribution of mass in the support of $\mathbb{P}_{\mathbf{r}}$.

In the first case, this is a contradiction since θ is optimized to minimize the distance between $d(\mathbb{P}_{\mathbf{r}}; \psi^*)$ and $d(\mathbb{P}_\theta; \psi^*)$. If the divergence between $\mathbb{P}_{\mathbf{f}}$ and $\mathbb{P}_{\mathbf{r}}$ increased, then we wouldn't observe a decrease in the divergence of the projection. The second case is also a contradiction since a sole mass redistribution of in the feature-space that increases divergence violates the assumption that a single infinitesimal optimization of θ that decreases the separation of $d(\mathbb{P}_{\mathbf{r}}; \psi^*)$ and $d(\mathbb{P}_\theta; \psi^*)$ □

Theorem 5.1 *GANs trained via Dynamic-GDA are Lyapunov stable. Also, in conditions where Sim-GDA and Alt-GDA are locally convergent, Dyn-GDA is too.*

Proof. (Sketch)

To prove GANs with Dynamic-GDA are Lyapunov stable, we need to show that for an arbitrary learning rate η, there exist finite upper-bounds on the distance (θ, ψ) will ever be from equilibrium.

To show this, we consider when Dynamic-GDA $c_1 = 1^+$ and $c_2 = 0^+$, then Dynamic-GDA will optimize ψ until a locally optimal discriminator ψ^* is achieved, or $\sup\{d(\mathbb{P}_\theta; \psi)\} < \inf\{d(\mathbb{P}_{\mathbf{r}}; \psi)\}$ for a bounded discriminator. If the discriminator is unbounded, such as in a WGAN, we halt the optimization of ψ once $\sup\{d(\mathbb{P}_\theta; \psi)\} < \inf\{d(\mathbb{P}_{\mathbf{r}}; \psi)\}$.

Importantly this logic is to help achieve GAN agnosticism, where Dyn-GDA can begin to optimize ψ until a "faithful" enough projection of the discriminator space is found in order to safely update θ

for at least 1 *iteration without promoting diverging dynamics, nor needing to know if* d *is bounded or not.*

Furthermore, as Mescheder et al. (2018) points out, when using neural works in the place of g *and* d*, there exists a set of equilibria due to the set of reparameterizations that produce identical functions. Thus, using assumptions* $I^{'}, II, III^{'}$ *in Mescheder et al. (2018) to consider training according to arbitrarily parameterized forms of neural networks* g *and* d*, so long as there exists* (θ^*, ψ^*) *in the space* $\boldsymbol{\theta} \times \boldsymbol{\psi}$*, we can show that Dyn-GDA is at least linearly convergent with infinitesimal learning rates locally around equilibria due to using the same set of update fields as Alt-GDA.* □

E Conventional Gradient Descent-Ascent Methods

Algorithm 3 Simultaneous Gradient Descent-Ascent

Require: (θ_0, ψ_0) initial parameters. $\mathcal{L}$, loss function. η, learning rate.
  while (θ, ψ) not converged **do**
    $\psi \leftarrow \psi + \eta \mathcal{L}(\theta, \psi)$
    $\theta \leftarrow \theta - \eta \mathcal{L}(\theta, \psi)$
  end while

Algorithm 4 Alternating Gradient Descent-Ascent

Require: (θ_0, ψ_0) initial parameters. $\mathcal{L}$, loss function. η, learning rate. n_d, number of discriminator updates per generator update.
  while (θ, ψ) not converged **do**
    for n_d times **do**
      $\psi \leftarrow \psi + \eta \mathcal{L}(\theta, \psi)$
    end for
    $\theta \leftarrow \theta - \eta \mathcal{L}(\theta, \psi)$
  end while

F Experimental Setup Details

We conduct all experiments on an RTX 4090, and use the PyTorch Paszke et al. (2019) framework. All experiments used the Adam optimizer for GANs with bounded discriminators, and RMSProp for GANs with unbounded discriminators. For GANs with regulation strategies baked into the architecture of the discriminator, Dynamic-GDA "turns off" these regularizations to decide the optimization to be conducted, and the gradient calculation for the updating uses the discriminator's regularization.

For experiments done pertaining to **celebA**, **CIFAR-10**, and **MNIST**, we train for 30,000 iterations, utilize a learning rate of $2 \cdot 10^{-4}$, a batch size of 128, and employ a 100-dimensional latent space sampled according to $\mathcal{N}(0, I)$. We normalize all images according to the distribution $\mathcal{N}(0.5, 0.5)$ for each channel. For Dynamic-GDA, we use $c_1 = 1.2, c_2 = 0.8$ on all image datasets.

For all $2D$-Datasets, we train for 30,000 iterations, utilize a learning rate of $2{\cdot}10^{-4}$, a batch size of 128, and employ a 10-dimensional latent space sampled according to $\mathcal{N}(0, I)$ as our $c_1 = 1.1, c_2 = 0.75$ for tuning Dynamic-GDA.

G ADDITIONAL VISUALIZATIONS

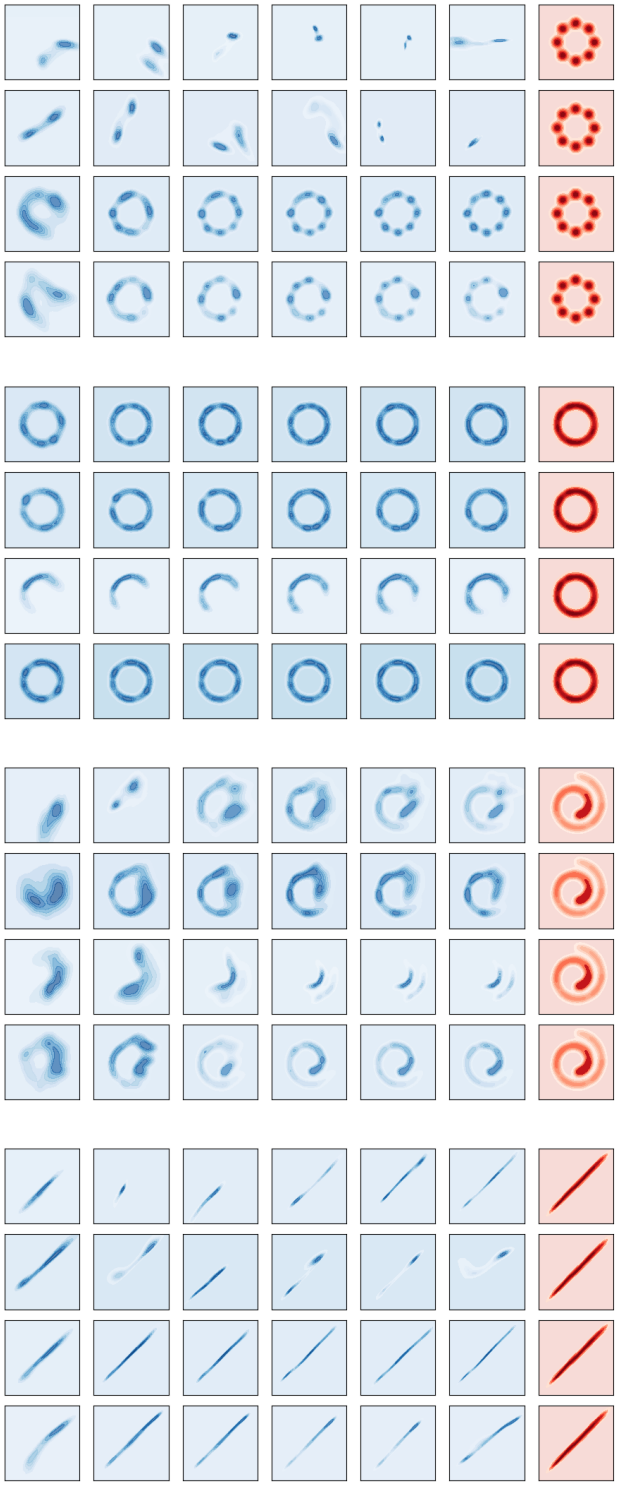

Figure 7: Distributions generated by BCE GANs at (from left to right) 5k, 10k, 15k, 20k, 25k, and 30k iterations when tasked with learning the target distributions (marked red) when equipped with either (from top to bottom) Sim-GDA, Alt-GDA$_{n_d=1}$, Alt-GDA$_{n_d=5}$, or Dyn-GDA. Grouped top-to-bottom are the Gaussian Ring, Circle, Spiral, and Line Segment datasets.

Figure 8: Distributions generated by GANs with JS-Regularization at (from left to right) 5k, 10k, 15k, 20k, 25k, and 30k iterations when tasked with learning the target distributions (marked red) when equipped with either (from top to bottom) Sim-GDA, Alt-GDA$_{n_d=1}$, Alt-GDA$_{n_d=5}$, or Dyn-GDA. Grouped top-to-bottom are the Gaussian Ring, Circle, Spiral, and Line Segment datasets.

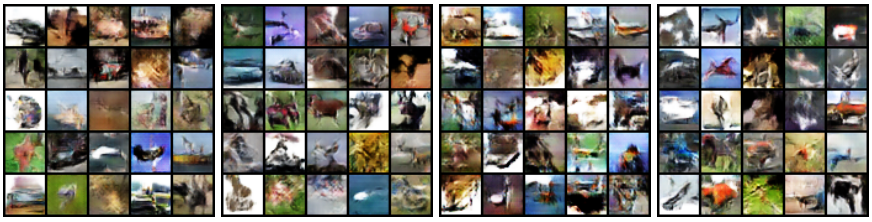

Figure 9: Synthetic samples generated after 30k iterations from Instance Noise GANs equipped with (from left to right) Sim-GDA, Alt-GDA$_{n_d=1}$, Alt-GDA$_{n_d=5}$, or Dyn-GDA during training.

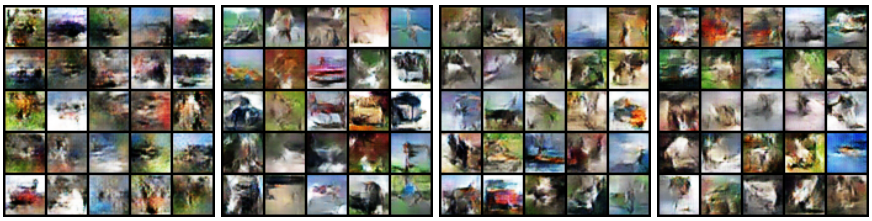

Figure 10: Synthetic samples generated after 30k iterations from WGANs equipped with (from left to right) Sim-GDA, Alt-GDA$_{n_d=1}$, Alt-GDA$_{n_d=5}$, or Dyn-GDA during training.

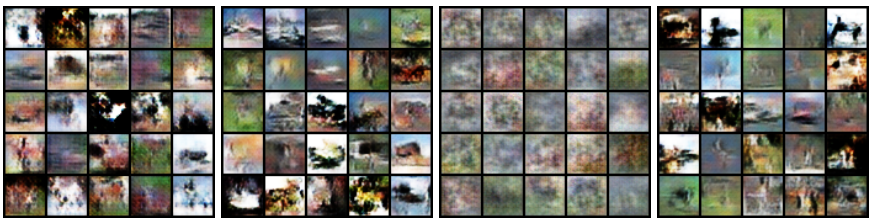

Figure 11: Synthetic samples generated after 30k iterations from GANs with Spectral Norm regularization equipped with (from left to right) Sim-GDA, Alt-GDA$_{n_d=1}$, Alt-GDA$_{n_d=5}$, or Dyn-GDA during training.

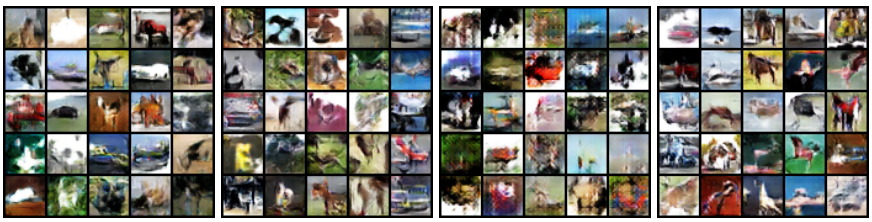

Figure 12: Synthetic samples generated after 30k iterations from GANs with JS-Regularization equipped with (from left to right) Sim-GDA, Alt-GDA$_{n_d=1}$, Alt-GDA$_{n_d=5}$, or Dyn-GDA during training.

Figure 13: Synthetic samples generated after 30k iterations from Instance Noise GANs equipped with (from left to right) Sim-GDA, Alt-GDA$_{n_d=1}$, Alt-GDA$_{n_d=5}$, or Dyn-GDA during training.

Figure 14: Synthetic samples generated after 30k iterations from WGANs equipped with (from left to right) Sim-GDA, Alt-GDA$_{n_d=1}$, Alt-GDA$_{n_d=5}$, or Dyn-GDA during training.

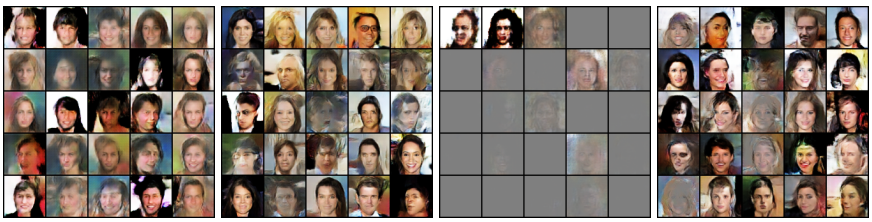

Figure 15: Synthetic samples generated after 30k iterations from GANs with Spectral Norm regularization equipped with (from left to right) Sim-GDA, Alt-GDA$_{n_d=1}$, Alt-GDA$_{n_d=5}$, or Dyn-GDA during training.

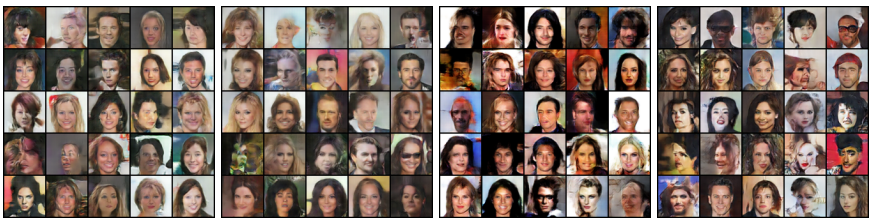

Figure 16: Synthetic samples generated after 30k iterations from GANs with JS-Regularization equipped with (from left to right) Sim-GDA, Alt-GDA$_{n_d=1}$, Alt-GDA$_{n_d=5}$, or Dyn-GDA during training.

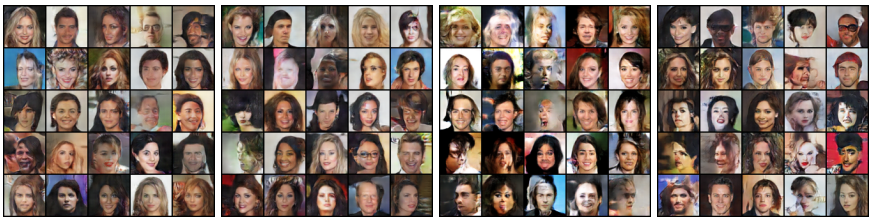

Figure 17: Synthetic samples generated after 30k iterations from GANs with R2-GP equipped with (from left to right) Sim-GDA, Alt-GDA$_{n_d=1}$, Alt-GDA$_{n_d=5}$, or Dyn-GDA during training.