On Duality Gap as a Measure for Monitoring GAN Training

Sahil Sidheekh , Aroof Aimen*, Vineet Madan*, Narayanan C. Krishnan
{2017csb1104, 2018csz0001, 2017csb1119, ckn} @iitrpr.ac.in
Indian Institute of Technology, Ropar

Abstract—Generative adversarial networks (GANs) are among the most popular deep learning models for learning complex data distributions. However, training a GAN is known to be a challenging task. This is often attributed to the lack of correlation between the training progress and the trajectory of the generator and discriminator losses and the need for the GAN’s subjective evaluation. A recently proposed measure inspired by game theory - the duality gap, aims to bridge this gap. However, as we demonstrate, the duality gap’s capability remains constrained due to limitations posed by its estimation process. This paper presents a theoretical understanding of this limitation and proposes a more dependable estimation process for the duality gap. At the crux of our approach is the idea that local perturbations can help agents in a zero-sum game escape non-Nash saddle points efficiently. Through exhaustive experimentation across GAN models and datasets, we establish the efficacy of our approach in capturing the GAN training progress with minimal increase to the computational complexity. Further, we show that our estimate, with its ability to identify model convergence/divergence, is a potential performance measure that can be used to tune the hyperparameters of a GAN.

I. INTRODUCTION

Generative Adversarial Network (GAN) [1] is probably one of the most important inventions in recent years that is widely popular for modeling data distributions, especially for high dimensional data such as images. A GAN aims to learn a data distribution under adversarial gameplay between two agents - a Discriminator (D) and a Generator (G) parameterized by \( \theta_d \) and \( \theta_g \) respectively. The discriminator (\( D : X \to [0,1] \)) differentiates between real data samples from the distribution to be learned and fake data samples generated by the generator, while the generator (\( G : Z \to X \)) aims to fool the discriminator; where \( X \) is the data space and \( Z \) is the low dimensional latent space. Formally, the GAN objective is defined as

\[
\min_{\theta_d \in \Theta_D} \max_{\theta_g \in \Theta_G} F(D,G),
\]

\[
F(D,G) = \mathbb{E}_{x \sim P_r}[\log(D(x))] + \mathbb{E}_{z \sim P_z}[\log(1 - D(G(z)))]
\]

where \( P_r \) is the real data distribution and \( P_z \) is a latent prior. The adversarial game using the above objective has been proven to minimize the Jensen-Shannon (JS) divergence between the true and generated data distributions.

Unlike classical machine learning tasks where the loss functions allow for direct inference on the training progress and model’s performance, GAN loss functions are non-intuitive and hard to interpret. This is partly due to the stochasticity involved in the training procedure, but mainly because the individual agents’ loss functions also depend on the adversary’s parameters. The alternate optimization changes the individual agents’ loss surface at every iteration, making inference of training progress a challenging task.

Recently, Grnarova et al. suggest the use of duality gap as a measure to monitor the training progress of a GAN. They establish through exhaustive experimentation and theoretical reasoning that the duality gap, being a domain agnostic measure can be used to evaluate GANs over a wide variety of tasks. While the true duality gap is an upper bound on the JS divergence between the real and generated distributions, estimating the true duality gap is an intractable problem. They suggest a gradient descent approach to approximate the duality gap. It is shown that the approximated duality gap can provide insights into the performance a GAN.

While the idea of using the duality gap is promising, the approach to estimating it has an inherent limitation due to the nature of gradient descent and the GAN loss surface. Gradient descent seeks first-order stationary points [2]. In a GAN, a first-order stationary point can be a Nash or non-Nash critical point. We show that the duality gap computed through gradient descent cannot distinguish between these two critical points, which is fundamental to GAN training. Thus, it is not useful for monitoring GAN training.

We propose an effective way to compute the duality gap using locally perturbed gradient descent. The central idea of the approach is to locally perturb a first-order stationary point before applying gradient descent. It can be shown that the local perturbation allows gradient descent to escape from a non-Nash critical point. This effectively differentiates the duality gap computed at Nash and non-Nash critical points.

Overall, we make the following contributions

• We demonstrate a problem with the prevalent approach to estimate the duality gap.
• We propose a theoretically grounded approach to estimate the duality gap through local perturbations that overcomes the limitations of the earlier approach.
• We conduct extensive experiments on a wide variety of GAN models and datasets to demonstrate the domain and model agnostic nature of the proposed measure to monitor the training process.
• We demonstrate that the duality gap can be used as a measure to influence the training process of a GAN, such as tuning hyperparameters. We also show that controllers

* Equal Contribution
that maintain the delicate balance between the generator and discriminator updates can be learned using rewards based on duality gap.

II. RELATED WORK

GANs have come a long way since it was proposed by [1]. The various challenges associated with a GAN such as convergence to non-Nash points, lack of diversity in generated samples, diminishing gradients, and enforcing optimal balance between the generator and discriminator have been well-studied. Several loss functions and model architectures have been proposed in this regard [3], [4], [5], [6], [7], [8], [9], [10], [11]. However, despite the numerous advantages, they still fail to provide an insight into the training progress of a GAN, owing to the non-intuitive nature of the associated loss curves. Thus, developing efficient measures for evaluating and monitoring GANs has attracted much interest in recent years [12], [13], [14].

While log-likelihood has been used traditionally to train and evaluate generative models, it is not a suitable measure for a likelihood-free model like GAN. Further, estimating likelihood is intractable and subject to error in higher dimensions [15]. Among the popular metrics to assess GANs are - Inception Score (IS) [16] and Frechet Inception Distance (FID) [9]. IS among the popular metrics to assess GANs are - Inception Score (IS) [16] and Frechet Inception Distance (FID) [9]. IS overestimation of manifold due to outliers.

estimating the probability density using $k$-means clustering to $k$-nearest neighbors.

Score (IS) [16] and Frechet Inception Distance (FID) [9]. IS is said to be a pure LNE, if:

Formally, a configuration or strategy $(\theta^*_g, \theta^*_d)$ is said to be a pure Nash equilibrium, if:

$$F(\theta^*_g, \theta^*_d) \leq F(\theta^*_g, \theta^*_d) \leq F(\theta^*_g, \theta^*_d) \forall \theta_g, \theta_d$$

(3)

Equivalently,

$$\min_{\theta_g \in \Theta_G} F(\theta_g, \theta^*_d) = \max_{\theta_d \in \Theta_D} F(\theta^*_g, \theta_d) = F(\theta^*_g, \theta^*_d)$$

(4)

However, finding the global Nash equilibrium is hard because the loss surface for the GAN optimization is not convex-concave [22]. Typically, a Local Nash Equilibrium (LNE) is what we expect to attain. Formally, a configuration $(\theta^*_g, \theta^*_d)$ is said to be a pure LNE, if $\exists \delta > 0$ such that $\forall \theta_g, \theta_d$ with $\|\theta_g - \theta^*_g\| < \delta$ and $\|\theta_d - \theta^*_d\| < \delta$:

$$F(\theta^*_g, \theta^*_d) \leq F(\theta^*_g, \theta^*_d) \leq F(\theta^*_g, \theta^*_d)$$

(5)

It is known that the GAN loss surface has abundant saddle points [8], [23]. Thus we must understand the characteristics of the LNE, a saddle point that we seek. As neural networks parameterize our models, we use gradient-based alternate optimization to seek the LNE $(\theta^*_g, \theta^*_d)$, which is a local maximum w.r.t the maximizing agent (discriminator) and a local minimum w.r.t the minimizing agent (generator). Formally, the local Nash Equilibrium $(\theta^*_g, \theta^*_d)$ satisfies

$$\nabla_{\theta_g} F(\theta^*_g, \theta^*_d) = 0$$

and

$$\nabla_{\theta_d} F(\theta^*_g, \theta^*_d) = 0$$

(6)

$$\frac{\partial^2 F(\theta^*_g, \theta^*_d)}{\partial \theta^*_g} > 0$$

and

$$\frac{\partial^2 F(\theta^*_g, \theta^*_d)}{\partial \theta^*_d} < 0,$$

(7)

We expect the gradient dynamics associated with the alternate optimization to enforce convergence to a local Nash Equilibrium. However, in practice, there are a plethora of non-Nash critical points that are attractors (satisfy only equation 6) under the gradient dynamics [8], thus making the training of GANs unstable and cumbersome.

B. Duality Gap for Monitoring GAN Training

A good measure for monitoring GAN training should be domain agnostic and enable easy inference of the training progress. Duality Gap [21] is a recently proposed measure motivated by principles of game theory.

Definition 1. (Duality Gap) Let $\Theta_D$ and $\Theta_G$ denote the parameter spaces of the discriminator and the generator respectively. Then for a pure strategy $(\theta_g, \theta_d)$ such that $\theta_g \in \Theta_G$ and $\theta_d \in \Theta_D$, the Duality Gap, is defined as
\[ DG(\theta_g, \theta_d) = \max_{\theta_d' \in \Theta_D} F(\theta_g, \theta_d') - \min_{\theta_g' \in \Theta_G} F(\theta_g', \theta_d) \] (8)

Intuitively, the duality gap measures the maximum payoff the agents can obtain by deviating from the current strategy. At Nash Equilibrium, as no agent can unilaterally increase their payoff, the duality gap is zero. This is implied by equation (4). Further, the duality gap is always non-negative. Gmarova et al. show that the duality gap is lower bounded by the Jensen-Shannon divergence between the true and generated data distributions. While the duality gap seems to be a useful performance measure for GANs, computing the exact duality gap at any point is an NP-hard problem as it involves finding the extrema of non-convex functions [24], [21].

1) Duality Gap Estimation: The duality gap is estimated through an iterative gradient based optimization [21]. An auxiliary generator and discriminator are initialized to the current values \((\theta_g^0, \theta_d^0)\) of their GAN counterparts at iteration \(t\). The auxiliary discriminator is optimized for a fixed number of iterations to obtain the worst case discriminator \((\theta_d^w)\) to compute \(M_1 = F(\theta_g^0, \theta_d^w)\). Analogously, the auxiliary generator is optimized to obtain the worst case generator \((\theta_g^w)\) to compute \(M_2 = F(\theta_g^w, \theta_d^0)\). The difference \(M_1 - M_2\) is the estimated duality gap at the iteration \(t\).

2) The Challenge: This estimation process seems very intuitive from the game theory perspective as it mimics the individual players’ efforts to unilaterally increase their payoffs by deviating from their current strategy. However, it limits the ability of the estimated duality gap as a performance metric to distinguish clearly between mode collapse, divergence and convergence encountered during GAN training. This is because, the gradient based optimizations for computing \(M_1\) and \(M_2\) seek first order stationary points i.e. points where the gradients of the objective function w.r.t. to the optimizing agent is zero. As Nash and non-Nash critical points are both first order stationary points [25], the duality gap estimated would be very close to zero. This is evident from the updates for estimating the worst case generator and discriminator. At the critical point \((\theta_g, \theta_d), \nabla_{\theta_g} F(\theta_g, \theta_d) = 0\) and \(\nabla_{\theta_d} F(\theta_g, \theta_d) = 0\) due to the first-order stationarity property. Thus it acts as an attractor in the gradient optimization for estimating the worst case generator \((\theta_g^w)\) and discriminator \((\theta_d^w)\). Thus \((\theta_g^w, \theta_d^w) = (\theta_g, \theta_d)\), which in turn results in \(M_1 = M_2\) and hence \(DG(\theta_g, \theta_d) = 0\). Thereby making it hard to differentiate between Nash and non-Nash critical points. However, at a non-Nash critical point, at least one of the agents can increase the payoff by deviating from the current strategy. It is the inherent limitation of the estimation process that the auxiliary models are unable to escape the non-Nash critical points [8].

To circumvent the optimization for estimating the worst discriminator and generator, [21] suggests estimating the approximate duality gap by choosing the most adversarial discriminator and generator from the saved set of snapshots of generator and discriminator parameters at different training time instants. This approximation requires the book-keeping of the discriminator and generator snapshots at different timestamps throughout the training. It is non-intuitive as it is uncertain if the worst generator/discriminator for a particular configuration would have been encountered previously during the GAN training.

C. Perturbed Duality Gap

We propose an effective way to compute the duality gap accurately using locally perturbed gradient descent. The method introduces perturbations to the worst-case agent’s initial point in its local neighborhood before applying gradient descent. Algorithm 1 summarizes the modified estimation process. The auxiliary generator and discriminators’ parameters are initialized to the current iterate values, \(\theta_g^t\) and \(\theta_d^t\), respectively. The initialization is perturbed by adding noise sampled uniformly from a ball of radius \(\sigma\). Optimization is performed on the perturbed initialization to estimate the worst-case generator \((\theta_g^w)\) and discriminator \((\theta_d^w)\). The success of the method relies on the perturbation that is applied to the initialization. We discuss the intuition of the local perturbation in the next subsection and present a way to limit the radius of the perturbation ball in the experiments.

Algorithm 1: Perturbed Duality Gap \(DG(\theta_g^t, \theta_d^t)\)

Input: Current iterate values of G and D - \(\theta_g^t, \theta_d^t\)

\[
\begin{align*}
\theta_g^t &\leftarrow \theta_g^t + \delta \\
\theta_d^t &\leftarrow \theta_d^t + \delta
\end{align*}
\]

where \(\delta \sim U([-\sigma, \sigma])\)

for \(i\) in Iterations do

\[
\begin{align*}
\theta_g^t &\leftarrow \theta_g^t + \eta \nabla_{\theta_g} F(\theta_g^t, \theta_d^t) \\
\theta_d^t &\leftarrow \theta_d^t - \eta \nabla_{\theta_d} F(\theta_g^t, \theta_d^t)
\end{align*}
\]

end

\((\theta_g^w, \theta_d^w)\) are the converged worst cases

\[
M_1 = F(\theta_g^t, \theta_d^w) \\
M_2 = F(\theta_g^w, \theta_d^t)
\]

return \(DG(\theta_g^t, \theta_d^t) = M_1 - M_2\)

1) Intuitive and Theoretical Understanding: The intuitive and theoretical understanding of the proposed method is inspired from the work on escaping saddle points during gradient descent [26]. Let \((\theta_g, \theta_d)\) be a configuration at which we would like to compute the duality gap. Without loss of generality, let us consider the optimization to compute \(M_2\). Let \((\theta_g^t, \theta_d^t)\) be the perturbed initial point and \((\theta_g^w, \theta_d^w)\) be the estimated worst case configuration. We can visualize \((\theta_g^t, \theta_d^t)\) as sampled from a perturbation ball \((B)\) centered at \((\theta_g, \theta_d)\) with radius \(\sigma\). The three possible situations at \((\theta_g, \theta_d)\) are (1) Gradients are sufficiently large; (2) Gradients are close to zero, but hessian is not positive semidefinite \((\nabla_{\theta_d} \sigma, \theta_d)\) is a first order stationary point); (3) Gradients are close to zero and hessian is positive semidefinite \((\nabla_{\theta_g} \sigma, \theta_g)\) is a second order stationary point). In the first case, as \(F(.)\) is smooth, classical gradient descent will reduce the function value.

The second and third cases are of specific interest because they are usually encountered during GAN training as convergence to non-Nash and Nash critical points respectively. If \((\theta_g, \theta_d)\) is a non-Nash critical point (the second case), we would like gradient descent starting from \((\theta_g^t, \theta_d^t)\) (the
perturbed initialization) to escape the saddle point thereby minimizing the function value. This is guaranteed by the theoretical results in [26] (Lemma 10). The intuition behind the theory revolves around understanding the geometry of the perturbation ball ($B$) around the saddle point. Let us denote by $B_S$ - the stuck region inside $B$ from which gradient descent would be unable to escape the saddle point. Jin et al. show that this region’s volume is minimal compared to the perturbation ball’s overall volume. Thus after adding the perturbation to $\theta$, the point $\theta'_g$ has a small chance of falling in $B_S$ and hence will escape from the saddle point efficiently. The same reasoning can be analogously applied to the discriminator parameters while computing $M_1$.

The direct inference from the above is that the perturbed duality gap estimate would be non-zero at non-Nash critical points and close to zero in the vicinity of Nash critical points, thus enabling better differentiation between convergence/non-convergence scenarios.

The local perturbations to the initial point might not be very intuitive from the game theory perspective as it does not precisely measure the effect of deviating from the current strategy. However, the estimation process is consistent because the perturbations are local (reachable) and only for the agent that is to be optimized. Hence should not significantly affect the optima to be attained as the loss surface remains unchanged. Incorporating this subtle modification will enable us to differentiate between convergence to Nash and non-Nash critical points.

We know that an LNE is a locally stable and attracting critical point for both the agents. Thus, despite the local perturbations, during convergence to a local Nash Equilibrium ($\theta^*_d, \theta^*_g$), the worst case discriminator while computing $M_1$ and the worst case generator while computing $M_2$ would ideally correspond to ($\theta^*_g$ and $\theta^*_d$) respectively. Thus the duality gap would still be close to zero.

When the agents converge to a non-Nash critical point ($\theta'_d, \theta'_g$), the original estimation process for $M_1$ and $M_2$ would result in the worst case discriminator and generator restricted to ($\theta'_d$ and $\theta'_g$) due to the lack of gradients in the vicinity of the saddle point. However, the introduction of local perturbations displaces the agent from the saddle point and provides the additional momentum required to escape the non Nash critical point, preventing the estimated duality gap from saturating to zero.

IV. EXPERIMENTS AND RESULTS

We design experiments to investigate the commonly encountered failure cases during GAN training from the perspective of the duality gap (DG). We empirically show that the proposed method, which we refer to as perturbed DG estimate, is better equipped to monitor the GAN training than the estimate of Grnarova et al., which we refer to as vanilla DG estimate. We also demonstrate how perturbed DG can be used to effectively influence the training of a GAN. Our objective is to only provide a rigorous comparison between the vanilla and perturbed DG estimate, for monitoring GAN training and is not on analyzing different GAN variants or datasets. The source code, the supplementary material, and other experimental details are publicly available 1.

A. Toy Game

Our first experiment is on a toy dataset where the Nash and non-Nash critical points can be easily visualized. This enables a clear understanding of the behavior of the two DG estimates. Consider a zero-sum game set in a 2-Dimensional space, between two agents - $x$ and $y$, guided by the following payoff function $f(x, y)$ well explored in optimization literature [8]

$$\min_{x} \max_{y} f(x, y) = e^{-0.01(x^2+y^2)((0.3x^2+y^2)^2+(x+0.5y)^2)^2}$$

Figure 1 depicts the contour plot of $f(.)$ together with its Nash (denoted as $A$ colored in green) and non-Nash (denoted as $B$ colored in red) critical points. In-order to estimate the duality gap at each of these points, we compute $M_1$ and $M_2$ by optimizing $f(.)$ w.r.t $x$ and $y$ respectively to obtain $x^w$ and $y^w$, using both vanilla and perturbed DG estimation processes. The optimization is performed using an Adam optimizer, with a learning rate of 5e-4 over 500 iterations. The radius of the perturbation ball is set to 0.01. The true duality gap would be zero at $A$ and positive at $B$. We observe (Figure1) that in the vicinity of $A$, $x^w$ and $y^w$ estimated using both vanilla and perturbed DG computation approaches converge to $A$, and the estimated duality gap is very close to zero, as expected. However, in the vicinity of $B$, $x^w$ and $y^w$ estimated using the vanilla approach remains stuck at $B$, resulting in the duality gap estimate becoming zero. On the other hand, $x^w$ estimated using our approach can escape the non-Nash critical point and reach a local optimum, resulting in a positive duality gap estimate value as expected at a non-Nash critical point.

1https://github.com/perturbed-dg/Perturbed-Duality-Gap
Figure 2: Perturbed and Vanilla DG estimate throughout the training progress of classic GAN (top row) and NSGAN (bottom row) on the RING and GRID datasets respectively for the scenarios - convergence (left column), mode collapse (middle column), and divergence (right column). For each subgraph, X-axis corresponds to the number of iterations, and Y-axis corresponds to the estimated duality gap.

An interesting observation here is that \( y^w \) estimated using our approach also converges to \( B \). This is expected because evaluating the hessian at \( B \) (see supplementary material), we see that \( B \) is locally unstable only for \( x \) and is a local optimum for \( y \). Thus, the duality gap estimated by our approach is consistent with the expected values, in contrast to the vanilla approach.

**B. Mixture of Gaussians**

We use three toy Gaussian mixture datasets (RING, GRID, and SPIRAL) to monitor the GAN training process using the estimated duality gap. For each of these datasets, we train a classical GAN and its non-saturating (NSGAN) counterpart. We simulate convergence, mode collapse, and divergence scenarios by varying the learning rate and update sequence of generator and discriminator. We analyze the results for training the classical GAN and NSGAN on the RING and GRID datasets. Similar patterns observed for the other datasets are discussed in the supplementary material.

Figure 2 demonstrates the sensitivity of the perturbed and vanilla duality gap estimates during convergence, mode collapse, and divergence throughout the training progress of a classic GAN and NSGAN on the RING and GRID datasets respectively. We observe that the behavior of perturbed and vanilla DG estimates is alike during convergence as both saturate close to zero as expected. The duality gap estimate being marginally negative at times despite the true duality gap being lower bounded by the JS divergence can be understood as a limitation posed by the approximation as discussed in [21], and does not to a large extent affect its ability to capture the training progress. We observed that the Vanilla DG estimate saturates to values very close to zero during mode collapse and divergence while perturbed DG saturates (or oscillates) to non-zero positive values as expected. Thus, we verify that our approach can escape non-Nash saddle points and, therefore, can better track the training progress of the GANs.

**C. Image Datasets**

Having established the efficacy of our approach on synthetic datasets, we now focus on confirming the generality of perturbed DG across GAN architectures in learning higher dimensional data distributions. To this end, we train a classical GAN, DCGAN, DCGAN+NS (NSGAN), WGAN-GP over image datasets - MNIST, Fashion-MNIST, CIFAR-10, and CelebA [27], [28], [29], [30]. To get an unbiased estimate of the DG, we split the datasets into a disjoint train, validation, and test sets to train the GAN, find the worst-case generator/discriminator, and evaluate the objective function w.r.t the worst-case agents respectively. We use classic GAN objective as a reference as suggested by [21] and train worst-case agents for 400 iterations using the optimizer of their GAN counterparts for estimating the duality gap. We simulate convergence and non-convergence scenarios by varying the learning rates.

1) **MNIST and Fashion MNIST**: Figure 3(a) validates the generality of perturbed DG estimates across GAN architectures for the MNIST dataset. As expected, we observe that both vanilla and perturbed DG estimates saturate close to zero during convergence irrespective of the GAN architecture. However, the vanilla DG estimate saturates close to zero even when the GAN diverges. On the contrary, the perturbed DG estimate shows a higher sensitivity to the divergence setting by saturating to a non-zero positive value. We repeat this experiment on the Fashion MNIST dataset and illustrate the results in the supplementary material. In addition to convergence and divergence settings, the perturbed DG estimate is also sensitive to mode collapse even for complex data distributions as suggested by the results presented in the supplementary material.

2) **CIFAR-10 and CelebA**: We investigate the generality of perturbed DG for monitoring the GAN training progress on complex image distributions like CIFAR-10 and CelebA. We choose WGAN-GP, a widely accepted GAN model for learning high dimensional data distributions, for this experiment. Results of this experiment (Figure 3(b)) show that the perturbed DG estimate can clearly differentiate between convergence and divergence, validating the hypothesis that the choice of the data distribution and GAN architectures do not constrain the sensitivity of the perturbed DG estimate towards convergence and (or) non-convergence.

**D. Influencing GAN Training using perturbed DG**

A generic framework for tuning the hyperparameters of a GAN with minimal human intervention has been an open problem in the community due to a lack of domain agnostic measures capable of accurately quantifying a GAN’s performance. The convergence of a GAN largely depends on preserving the delicate balance between the generator and discriminator [11] that is governed by the tuning of hyperparameters. We show empirically that perturbed DG can be used to fine tune these hyperparameters. Further, perturbed DG can facilitate
learning of meta-models that can automatically drive a GAN to convergence.

1) **Hyperparameter Search:** We focus on tuning the learning rates and model capacity. We perform a grid search on these hyperparameters and compute the perturbed DG estimate at every choice to identify the optima. The learning rates of both the generator and the discriminator impact each other. Thus we search over the 2D space defined by both the parameters. Figure 4(a) depicts the contour plot for the perturbed DG estimate over this space for a classic GAN trained on the RING dataset. The optimal range of learning rates for the generator and the discriminator can be easily identified as the region where perturbed DG approaches zero. The optimal learning rates, as suggested by the perturbed DG estimate, are further verified by visualizing the similarity of the generated and true data distribution in these regions. As for model capacity, we search over the space defined by the number of hidden nodes in each layer of the models - we use the same number of layers and hidden nodes for both the generator and the discriminator. This is visualized in Figure 4(b) for a classical GAN trained on the GRID dataset. We observe an optimum capacity for the models at which the game is balanced, and the perturbed DG estimate can identify it as the region where the duality gap is close to zero.

2) **Dynamic Scheduling:** Another dimension governing the balance between the generator and discriminator is the schedule that drives the underlying alternate iterative optimization. [31] shows that a dynamic schedule that considers the current state and the history of the optimization procedure over following a fixed schedule can lead to better convergence for GANs. They use a controller network, learned using a policy gradient method to maximize the expected reward defined in terms of the final performance of the GAN, to generate a dynamic schedule. However, they quantify the GAN performance using the Inception Score, limiting its applicability to only images. We extend their approach by superseding the inception score with perturbed DG as the reward for training the controller. We define the input space of the controller as a 4-tuple - (a) the log-ratio of the magnitude of the gradients of the generator and discriminator, an exponential moving average of (b) generator’s loss, (c) discriminator’s loss, and (d) perturbed DG. Such a state representation is independent of the GAN architecture and captures sufficient information needed to infer the balance between the agents. As there is an inverse relation between perturbed DG ($DG$) and convergence of a GAN, we specify the reward as $\alpha DG + \epsilon$, where $\alpha$ is a reward constant and $\epsilon$ is added for numerical stability. The domain agnostic nature of the duality gap enables learning dynamic schedules for GANs irrespective of the data distribution and also facilitates learning a domain agnostic controller that is generalizable across GAN architectures and domains. We verify this hypothesis by training a controller using perturbed DG to enforce convergence of a WGAN-GP on the 2D RING dataset. We then use this trained controller to drive the training of a WGAN-GP on the 2D GRID dataset and a convolutional WGAN-GP on the MNIST dataset.

The implementation details regarding the training procedure of the controller are provided in the supplementary material. Figure 5 depicts the average performance of the dynamic schedule against the suggested discriminator-generator update ratio of 5:1 for WGAN-GP, in terms of domain-specific evaluation criteria for each the above settings - KL divergence for 2D GRID and Inception Score for MNIST. We observe that
the dynamic schedule leads to faster and better convergence for each of the above settings. The KL divergence between the real and generated distributions quickly saturates close to zero for the 2D GRID and the higher inception scores for MNIST are obtained when using the dynamic schedule predicted by the controller network. Thus, while the balance enforced by the controller helps transcend manual tuning of the schedule, its generalizability, owing to the domain agnostic nature of perturbed DG, outweighs the effort invested in training.

E. Ablation Studies

1) Perturbation Ball Radius: An unknown in the proposed perturbed DG estimation process is $\sigma$, the radius of the perturbation ball. We investigate the effect of $\sigma$ on DG estimate for three settings, namely convergence, mode collapse, and divergence. We train a classic GAN on the RING dataset for conducting this study. We estimate the saturated DG value at the end of 20,000 iterations for each setting and monitor the impact of $\sigma$ on the perturbed DG values across 50 trials. We observe from figure 6(a) that the DG steadily increases with $\sigma$ during the convergence setting. However, there is a steeper increase in DG values for mode collapse and divergence settings, saturating to a positive value. The size of the stuck region in the perturbation ball could explain this behavior. During convergence, the model converges to a Nash point enclosed in a larger stuck region within the perturbation ball due to which sufficiently large $\sigma$ is required to push the model off the stuck region. On the contrary, during non-convergence, the model converges to a non-Nash critical point enclosed in a comparatively thin stuck region. Therefore a small increase in $\sigma$ is sufficient to push the model off the stuck region.

We also notice an increasing trend in the DG variance for the convergence setting across trials compared to other scenarios. This behavior is explained by observing that the perturbations across the trials are stochastic, and for some trials, it may be sufficient for the model to escape the stuck region compared to other trials. However, a small perturbation is sufficient for the model to escape the stuck region during mode collapse and divergence, indicating that the stuck region’s volume is small at these non-Nash critical points arrived during GAN training.

We now draw the attention towards the approximation of $\sigma$ by the standard deviation of the model’s weights. In figure 6(a), red, blue, and green vertical lines correspond to the average standard deviation of the model’s weights for convergence, mode collapse, and divergence settings, respectively. It is evident from the figure that for perturbation ball radii comparable to these standard deviations, perturbed DG is able to differentiate the corresponding scenarios.

2) Computational Complexity: The two contributing factors for an increase in the computational complexity while monitoring GAN training using perturbed DG are - number of training iterations of the auxiliary model and the frequency of perturbed DG computation. We investigate the additional cost imposed by the DG estimation process under these two factors. We train a classical GAN on the RING dataset. In the first experiment, we track the estimated DG values as a function of auxiliary GAN’s training iterations. Figure 6(b) shows the trend in the perturbed DG values. It is evident from the figure that less than 200 iterations are sufficient for the convergence of the DG for various settings.

Figure 7(a) presents the increase in the computational time (averaged over 5 trials) when estimating the DG at various intervals. The number of iterations for training the auxiliary GAN is fixed at 200 as suggested by the early stopping criteria. It can be seen that for higher computation intervals, which are quite practical to work with, the computation time is comparable to the GAN training process without any DG computation. We show that this higher DG computation interval does not degrade the trend captured by the duality gap in figure 7(b). The variance in the DG estimates across the iterations for varying estimation.
frequency is negligible.

V. CONCLUSION

Evaluating the training progress of a GAN has been an open problem in the machine learning community. Recently Grnarova et al. establish that the true duality gap is an upper bound on the JS divergence between the real and generated distributions, and through exhaustive experimentation, demonstrate that the duality gap is domain agnostic and can be used to evaluate GANs over a wide variety of tasks. Computing the true duality gap for a GAN is an intractable problem, and thus Grnarova et al. suggest a gradient descent approach to estimate its approximation. This paper shows that the duality gap estimation process suffers from an inherent limitation of being unable to distinguish between Nash and non-Nash attractors of the gradient dynamics associated with the GAN optimization. We present a thorough study of this limitation and propose a method to overcome it without increasing the computational complexity, through local perturbations. Through rigorous experiments, we demonstrate the usefulness of the perturbed duality gap for monitoring GAN training progress. We also show that it is possible to learn generic meta-models capable of driving GANs to convergence using the perturbed duality gap. This potentially opens up new possibilities for automated GAN training.

VI. ACKNOWLEDGEMENTS

The resources provided by ‘PARAM Shivay Facility’ under the National Supercomputing Mission, Government of India at the Indian Institute of Technology, Varanasi and under Google Tensorflow Research award are gratefully acknowledged.

REFERENCES

[1] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, “Generative adversarial nets,” in Advances in Neural Information Processing Systems, 2014.

[2] P. Jain and P. Kar, “Non-convex optimization for machine learning,” Foundations and Trends in Machine Learning, 2017.

[3] C.-L. Li, W.-C. Chang, Y. Cheng, Y. Yang, and B. Póczos, “Mind gan: Towards deeper understanding of moment matching network,” in Advances in Neural Information Processing Systems, 2017.

[4] M. Arjovsky, S. Chintala, and L. Bottou, “Wasserstein generative adversarial networks,” in International Conference on Machine Learning, 2017.

[5] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville, “Improved training of wasserstein gans,” in Advances in Neural Information Processing Systems, 2017.

[6] Y. Qin, N. J. Mitra, and P. Wonka, “How does lipschitz regularization influence GAN training?” in European Conference on Computer Vision, 2020.

[7] T. Miyato, T. Kataoka, M. Koyama, and Y. Yoshida, “Spectral normalization for generative adversarial networks,” in International Conference on Learning Representations, 2018.

[8] 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, 2019.

[9] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, “Gans trained by a two time-scale update rule converge to a local nash equilibrium,” in Advances in Neural Information Processing Systems, 2017.

[10] A. Radford, L. Metz, and S. Chintala, “Unsupervised representation learning with deep convolutional generative adversarial networks,” in International Conference on Learning Representations, 2016.

[11] D. Zhang and A. Khoreva, “Progressive augmentation of gans,” in Advances in Neural Information Processing Systems, 2019.

[12] A. Borji, “Pros and cons of GAN evaluation measures,” arXiv preprint, 2018.

[13] M. Lucic, K. Kurach, M. Michalski, S. Gelly, and O. Bousquet, “Are gans created equal? a large-scale study,” in Advances in Neural Information Processing Systems, 2018.

[14] S. H. Olsson, D. Blupatiraju, T. B. Brown, A. Odena, and I. J. Goodfellow, “Skill rating for generative models,” arXiv preprint, 2018.

[15] L. Theis, A. van den Oord, and M. Bethge, “A note on the evaluation of generative models,” in International Conference on Learning Representations, 2016.

[16] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, “Improved techniques for training gans,” in Advances in Neural Information Processing Systems, 2016.

[17] M. S. Sajjadi, O. Bachem, M. Lucic, O. Bousquet, and S. Gelly, “Assessing generative models via precision and recall,” in Advances in Neural Information Processing Systems, 2018.

[18] T. Kynkänniemi, T. Karras, S. Laine, J. Lehtinen, and T. Aila, “Improved precision and recall metric for assessing generative models,” in Advances in Neural Information Processing Systems, 2019.

[19] I. O. Tolstikhin, S. Gelly, O. Bousquet, C.-J. Simon-Gabriel, and B. Schölkopf, “Adagan: Boosting generative models,” in Advances in Neural Information Processing Systems, 2017.

[20] M. F. Naeem, S. J. Oh, Y. Uh, Y. Choi, and J. Yoo, “Reliable fidelity and diversity metrics for generative models,” arXiv preprint, 2020.

[21] P. Grnarova, K. Y. Levy, A. Lucchi, N. Perraudin, I. Goodfellow, T. Hofmann, and A. Krause, “A domain agnostic measure for monitoring and evaluating gans,” in Advances in Neural Information Processing Systems, 2019.

[22] C. Jin, P. Netrapalli, and M. I. Jordan, “What is local optimality in nonconvex-nonconcave minimax optimization?” arXiv preprint, 2019.

[23] A. Choromanska, M. Henaff, M. Mathieu, G. B. Arous, and Y. LeCun, “The Loss Surfaces of Multilayer Networks,” in Artificial intelligence and statistics, 2015.

[24] K. K. Thekumparampil, P. Jain, P. Netrapalli, and S. Oh, “Efficient algorithms for smooth minimax optimization,” in Advances in Neural Information Processing Systems, 2019.

[25] T. Fiez, B. Chasnov, and L. J. Ratliff, “Convergence of learning dynamics in stackelberg games,” arXiv preprint, 2019.

[26] C. Jin, R. Ge, P. Netrapalli, S. M. Kakade, and M. I. Jordan, “How to escape saddle points efficiently,” in International Conference on Machine Learning, 2017.

[27] L. Deng, “The mnist database of handwritten digit images for machine learning research,” 2012.

[28] H. Xia, K. Rasul, and R. Vollgraf, “Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms,” arXiv preprint, 2017.

[29] A. Krizhevsky, V. Nair, and G. Hinton, “Cifar-10 (canadian institute for advanced research),” 2014.

[30] Z. Liu, P. Luo, X. Wang, and X. Tang, “Deep learning face attributes in the wild,” in International Conference on Computer Vision, 2015.

[31] H. Xu, H. Zhang, Z. Hu, X. Liang, R. Salakhutdinov, and E. P. Xing, “Autoloss: Learning discrete schedule for alternate optimization,” in International Conference on Learning Representations, 2019.