# Supplementary Material - On Duality Gap as a Measure for Monitoring GAN Training

## Overview

This section details the experimental setup and observations left out in the main paper due to space constraints. The supplementary material is organized as follows:

- **GAN losses are non-intuitive:** We illustrate situations often encountered during GAN training, where the non-intuitive nature of the GAN loss curves are apparent. This motivates the need for a reliable metric for monitoring GAN training.
- **Toy function:** This section illustrates the behavior of perturbed and vanilla DG estimates in the vicinity of Nash and non-Nash critical points of a toy function.
- **Synthetic 2D datasets:** In this section, we compare the performance of vanilla and perturbed DG estimate during the training of various GANs (classical and NSGAN) on 2D synthetic datasets (RING, GRID, SPIRAL).
- **Image datasets:** The comparative analysis of vanilla and perturbed DG on high dimensional image datasets is presented in this section.
- **Dynamic Scheduling Using Perturbed DG:** The implementation details and observations pertaining the training of a meta-model to drive a GAN convergence are presented in this section.
- **Ablation studies:** The last section analyzes the trend of the perturbed DG during GAN training w.r.t the radius of the perturbation ball $\sigma$.

## Non-Intuitive Nature of GAN Loss Curves

With the advent of deep learning, most machine learning tasks have been reduced to function optimization problems, popularly solved through gradient descent iteratively. The surface of the optimization objective is hard to visualize due to the large number of parameters. However, for classical machine learning tasks, it is usually possible to infer whether the model has converged by analyzing the trend in the values of the objective function that is being optimized. We would ideally expect the loss curves associated with the model to decrease and eventually saturate, indicating convergence. Such intuitive inferences, however, cannot be made for a GAN. This is because GANs involve, in addition to the stochasticity, an alternating iterative optimization where the loss surfaces of the generator and discriminator change at every iteration. Thus, while we do expect the divergence between the real and generated distributions to eventually decrease, the losses of the individual models need not always decrease steadily to result in convergence (Fedus et al. 2018). This is also implied by the adversarial nature of the GAN game - as the objectives of the individual agents are conflicting, an equilibrium cannot be attained through consistent reduction in the losses of both the models. Figure 1 shows the loss curves of a classic GAN trained over the RING dataset across different settings. We observe that during convergence, the discriminator's loss decreases, while the generator's loss increases where-after they saturate. However, similar behaviour is also observed in the loss curves during a stable mode collapse and divergence. In fact, during divergence, despite the slight variance, the average loss of the models is much lower than that corresponding to the convergence scenario. Thus, analyzing the performance of a GAN from its loss curves is a cumbersome task. This motivates the need for better measures to quantitatively evaluate and infer the learning of a GAN.

## Toy Game

### Theoretical Verification

Consider a zero-sum game set in a 2-Dimensional space, between two agents - $x$ and $y$, guided by the following payoff function $f(x, y)$

$$\min_y \max_x f(x,y) = e^{-0.01(x^2+y^2)}((0.3x^2+y)^2+(x+0.5y^2)^2) \quad (1)$$

We verify the dynamics of the mini-max game defined by Equation 1 in the vicinity of the critical points depicted in figure **??**.

Let $A = (-12.43373, -8.78737)$ and $B = (0.0, 0.0)$ . We have,

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

Evaluating the gradient and hessian at $B$,

$$\nabla_x f(B) = 0.0 \;\; \& \;\; \nabla_y f(B) = 0.0$$
$$\nabla^2_{xx} f(B) = 2.0 > 0 \;\; \& \;\; \nabla^2_{yy} f(B) = 2.0 > 0$$

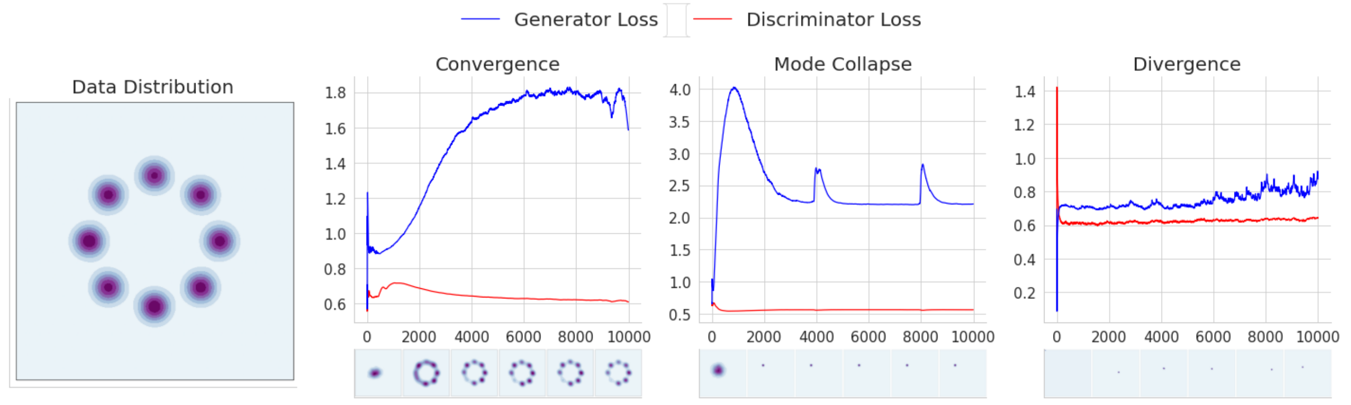


Figure 1: Generator and discriminator losses throughout the training progress of classic GAN on RING dataset for convergence, divergence and mode collapse settings.

Thus $B$ is a local minima w.r.t $y$, but is not a local maxima w.r.t $x$. Clearly, $B$ is a Non-Nash critical point.
Estimating the duality gap at $B$, we have :

Vanilla DG Estimate = 0.00
Perturbed DG Estimate = 49.217

Evaluating the gradients and hessian at $A$,

$$\nabla_x f(A) = 0.06 \;\; \& \;\; \nabla_y f(A) = -0.06$$
$$\nabla^2_{xx} f(A) = -1.1 < 0 \;\; \& \;\; \nabla^2_{yy} f(A) = 9.8 > 0$$

Thus, $A$ is both a local minima w.r.t $y$, and a local maxima w.r.t $x$. $A$ is thus a Nash critical point. Estimating the duality gap at $A$, we have :

Vanilla DG Estimate = 0.001
Perturbed DG Estimate = 0.002

We thus verify that our approach is able to differentiate between convergence to Nash and Non Nash critical points more effectively than the vanilla duality gap estimation approach.

## Synthetic 2D datasets

The sensitivity of vanilla and perturbed DG to convergence and non-convergence scenarios during the training procedure of a classic GAN and a non-saturating (NS)GAN across the 2D synthetic datasets- RING, GRID and SPIRAL are presented in figures 2 and 3 respectively. We observe that almost in all cases of mode collapse and divergence, perturbed DG shows better sensitivity than vanilla DG by saturating (or oscillating) to a non-zero positive values.

### Implementation Details

While training the classical GAN, we fix the learning rate as 5e-4 for both the generator and the discriminator across all scenarios. To simulate convergence, stable mode collapse and divergence, we use a discriminator-generator update ratio of 1:1, 15:1, and 1:15 respectively. While training NS-GAN, we use learning rates as 1e-3, 1e-4, 1e-3, and update ratios as 3:2, 5:7, 1:10 respectively for convergence, mode collapse, and divergence scenarios. The learning rate is kept the same for both the generator and the discriminator. We use a latent noise dimension of 100 for the generator. The architectural details of the models are summarized in Table-1. For the computation of the duality gap, we train the individual models using Adam optimizer for 300 iterations to find the worst-case generator/discriminator. The local perturbations added to the individual weight layers are drawn from a uniform distribution having standard deviation equal to twice the standard deviation of the weights of the corresponding layers.

| | Generator | Discriminator |
|---|---|---|
| Architecture | dense(128), relu<br>dense(128), relu<br>dense(2) | dense(128), relu<br>dense(128), relu<br>dense(1), sigmoid |
| optimizer | Adam | Adam |

Table 1: Model Details - 2D Experiments

## Image datasets

### MNIST and Fashion MNIST

**Convergence and Divergence on Fashion MNIST** To show the generality of perturbed DG across datasets, we compare perturbed and vanilla DG estimates during the training progress of various GAN's on the Fashion MNIST dataset. Figure 4 shows the results obtained. We observe that perturbed DG is more sensitive to divergence setting than vanilla DG irrespective of the GAN type. Therefore our presumption that perturbed DG is robust across datasets and GAN architecture is valid.

**Implementation Details** The architecture and hyper-parameter details of different GANs for convergence and divergence settings on MNIST and Fashion MNIST data-sets are discussed here. For classical GAN, we use fully connected layers followed by leaky relu activation with alpha = 0.3. To simulate convergence and divergence, we use 3 hidden layers with 512 nodes each, for both generator and discriminator, followed by a projection layer. For DCGAN,

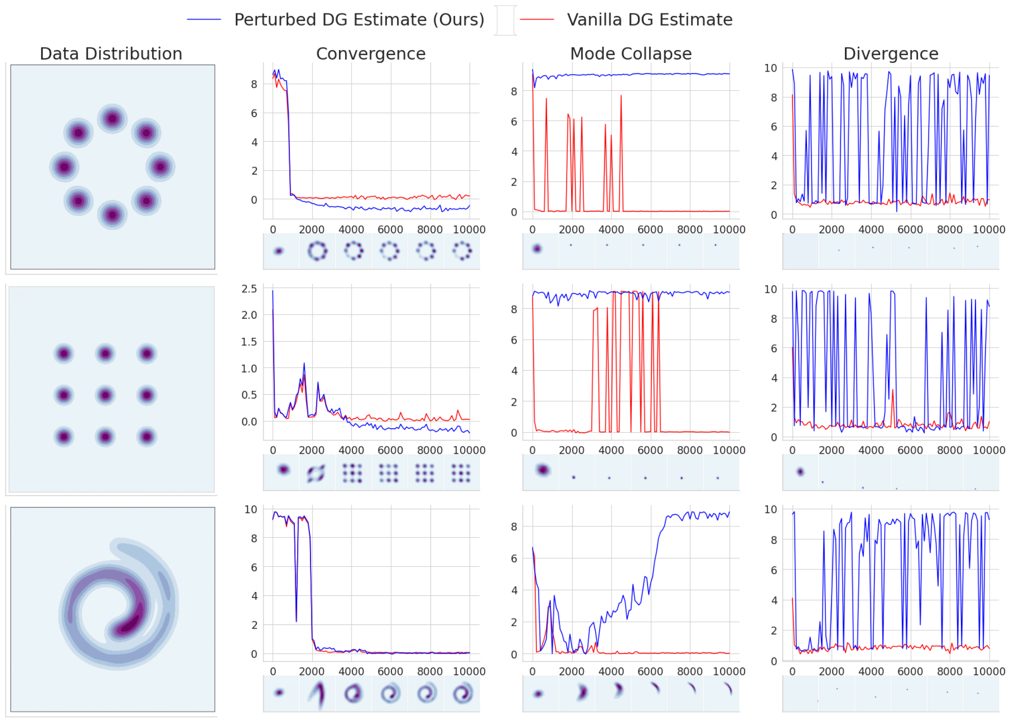


Figure 2: Perturbed and Vanilla DG estimate throughout the training progress of classic GAN on 2D data-sets for convergence, mode collapse, and divergence.

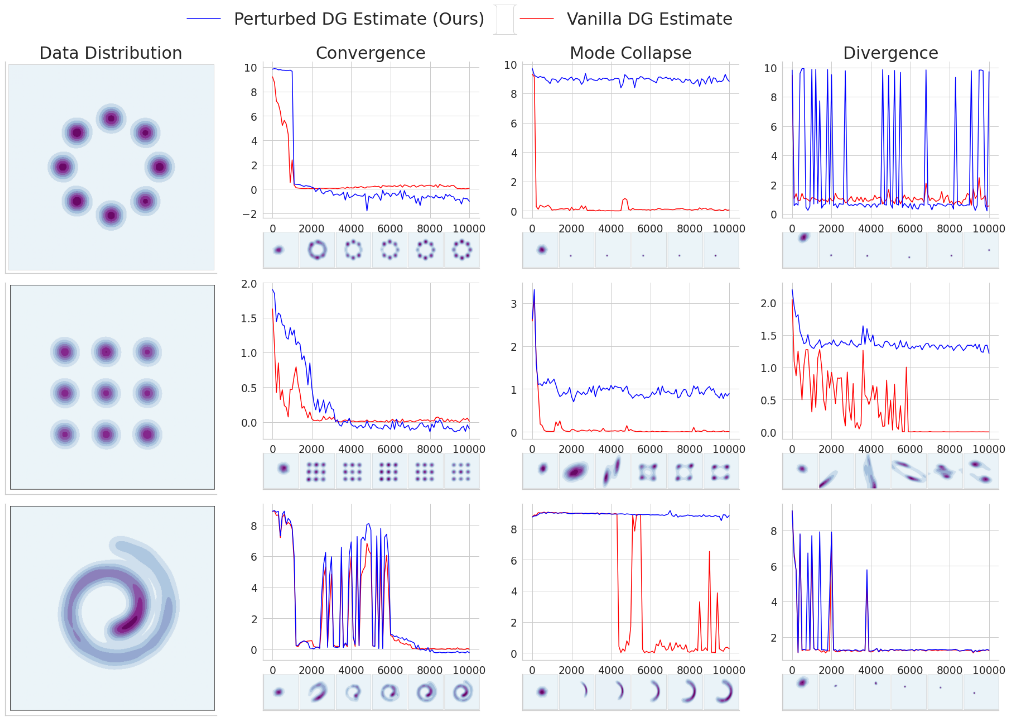


Figure 3: Perturbed and Vanilla DG estimate throughout the training progress of NSGAN on 2D data-sets for convergence, mode collapse, and divergence.

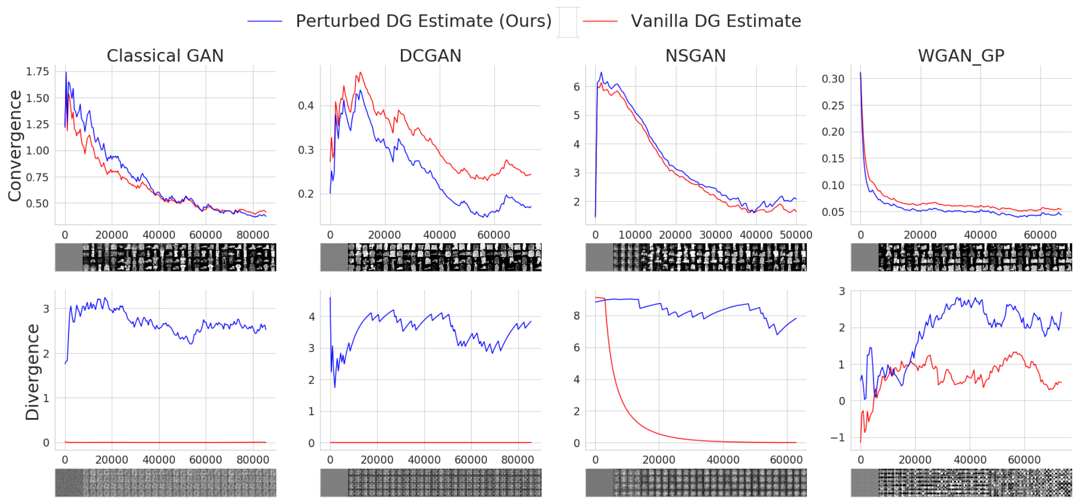


Figure 4: Comparison of perturbed and vanilla DG estimate throughout the training progress of classic GAN (Col-1), DCGAN (Col-2), NSGAN (Col-3) and WGAN-GP (Col-4) on Fashion MNIST dataset for convergence (Row-1) and divergence (Row-2) settings. For each sub-graph, X-axis corresponds to the number of iterations, and Y-axis corresponds to DG values.

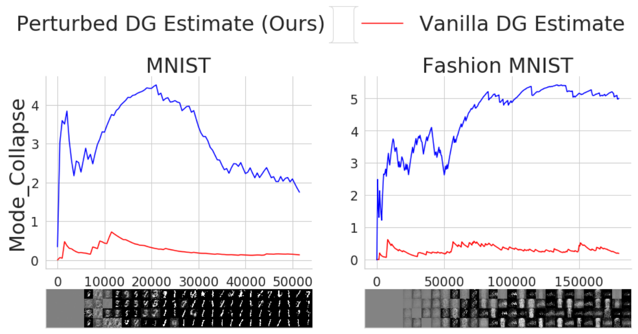


Figure 5: Comparison of perturbed and vanilla DG estimate obtained during the training progress of classic GAN on MNIST (Col-1) and Fashion MNIST (Col-2) datasets for mode collapse setting.

|  | GAN_Type | G_lr | D_lr | $\sigma$ |
|---|---|---|---|---|
| Convergence | Classic GAN | 1e-4 | 1e-4 | 0.01 |
|  | DCGAN | 2e-4 | 1e-4 | 0.01 |
|  | NSGAN | 2e-5 | 1e-5 | 0.01 |
|  | WGAN-GP | 5e-5 | 5e-5 | 0.2 |
| Divergence | Classic GAN | 2e-3 | 4e-3 | 0.1 |
|  | DCGAN | 1e-4 | 6e-4 | 0.1 |
|  | NSGAN | 7e-5 | 1e-5 | 0.05 |
|  | WGAN-GP | 1e-4 | 1e-2 | 0.5 |

(a)

|  | GAN_Type | G_lr | D_lr | $\sigma$ |
|---|---|---|---|---|
| Convergence | Classic GAN | 1e-4 | 1e-4 | 0.01 |
|  | DCGAN | 2e-4 | 1e-4 | 0.01 |
|  | NSGAN | 2e-5 | 1e-5 | 0.01 |
|  | WGAN-GP | 5e-5 | 5e-5 | 0.2 |
| Divergence | Classic GAN | 2e-3 | 4e-3 | 0.1 |
|  | DCGAN | 1e-4 | 6e-4 | 0.1 |
|  | NSGAN | 7e-5 | 1e-5 | 0.05 |
|  | WGAN-GP | 1e-4 | 1e-2 | 0.5 |

(b)

Table 2: Hyper-parameter Details -(a) MNIST Experiments (b) Fashion MNIST Experiments

NSGAN, and WGAN-GP, we use convolutional layers followed by batch normalization and relu activation function. We project the latent noise to $7 \times 7 \times 256$ and use three transpose convolution layers containing 128, 64, 1 filters with strides 1, 2, 2, respectively for the generator. For the discriminator, we use two convolution layers containing 64, 64 filters with strides 2, 2 respectively, followed by the projection layer. The convolutional filter size is set to $5 \times 5$. The latent noise is of dimension 100. We fix the update ratio as 1:1 and use Adam optimizer across all settings for all GANs. The learning rates and the radius of the perturbation balls' (for computing DG) for convergence and divergence settings for different GANs on MNIST and Fashion MNIST datasets are given in Table 2(a) and 2(b) respectively.

**Mode Collapse on MNIST and Fashion MNIST** In this experiment, our goal is to show that the sensitivity of the perturbed duality gap towards mode collapse setting is robust across datasets. It is challenging to simulate mode collapse in complex GAN architectures, so we choose classical GAN on MNIST and Fashion MNIST datasets for this experiment. We simulate mode collapse by tuning the hyperparameters (Table 3) and architecture of classic GAN. The architecture comprises of fully connected layers followed by leaky relu activation with alpha = 0.3. We use 5 hidden layers with 256 nodes each, for generator and 4 hidden layers with 256 nodes each, for the discriminator.

|  | G_lr | D_lr | $\sigma$ |
|---|---|---|---|
| MNIST | 1e-4 | 5e-4 | 0.1 |
| Fashion MNIST | 5e-5 | 5e-4 | 0.1 |

Table 3: Hyper-parameter details of classic GAN for mode collapse setting

|  | Dataset | G_lr | D_lr | $\sigma$ |
|---|---|---|---|---|
| Convergence | CIFAR10 | 1e-4 | 1e-4 | 1e-2 |
|  | CelebA | 2e-4 | 1e-4 | 1e-2 |
| Divergence | CIFAR10 | 1e-4 | 1e-2 | 5e-1 |
|  | CelebA | 5e-5 | 5e-1 | 5e-1 |

Table 4: Hyper-parameter Details - WGAN-GP on CIFAR10 and CelebA Experiments.

The results of this experiment are illustrated in figure 5. We observe that the perturbed DG is more sensitive to the mode collapse setting than vanilla DG as it saturates to a positive value, unlike vanilla DG, which is close to zero.

### CIFAR-10 and CelebA

**Implementation Details** We use WGAN-GP for conducting experiments on CIFAR-10 and CelebA datasets owing to its training stability. We project the latent noise to $4 \times 4 \times 256$ and use four transpose convolution layers containing 64, 64, 64, 3 filters with strides 2, 2, 1, 2 respectively for the generator. For the discriminator, we use three convolution layers containing 64, 64, 64 filters with strides 2, 2, 2 respectively, followed by the projection layer. Also, a standard discriminator to generator update ratio of 5:1 is used. Other implementation details of WGAN-GP is the same as discussed under MNIST and Fashion MNIST datasets. The hyper-parameter details are given in Table 4.

## Dynamic Scheduling Using Perturbed DG

Following the approach of (Xu et al. 2019), we use reinforcement learning framework to train a controller to guide the GAN optimization. The state space comprises of 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. The controller learns to output the posterior probability of choosing the agents (Generator and Discriminator) given the current state of optimization. The agent to be optimized at each iteration is sampled using the probability distribution defined by the contoller's output. Since such a sampling is non-differentiable, the parameters of the controller are learned using the REINFORCE (Williams 1992) algorithm, where the reward for the controller's action is defined as the downstream GAN performance in terms of perturbed DG. More specifically, we define the reward as $\frac{\alpha}{DG+\epsilon}$ where $DG$ is the exponential moving average of perturbed DG at the end of a training episode and $\epsilon$ is added for numerical stability. An episode consists of $K$ optimization steps for the GAN, guided by the controller. On termination of an episode, the expected reward corresponding to each action in the dynamic schedule generated by the controller is computed and the controller is updated to maximize this reward. An episode may also terminate if the output of the controller collapses to the same action, during which a negative reward is provided.

We train the controller to guide the optimization of a WGAN-GP on the RING dataset. We test this controller on the GRID and SPIRAL datasets. Figure 6 compares the performance of the GAN using the dynamic schedule given by the controller as opposed to a fixed schedule. We consider two fixed (D:G) update ratios - 5:1 and 1:5. We observe that though the standard 5:1 fixed ratio eventually leads to convergence for both GRID and SPIRAL, using the dynamic schedule leads to faster convergence. This is evident from the KL divergence plots and the visualization of the generated distribution across iterations. On the other hand, using a fixed update ratio of 1:5 leads to divergence. Thus, the controller trained using perturbed DG helps supersede the effort that manual tuning of the update ratio demands.

Figure 7 represents the selection frequency of the generator and discriminator using the dynamic schedule within intervals of varying resolution across the training iterations of a WGAN-GP on the GRID dataset. We observe that the variance in the selection frequencies is lower within intervals of higher resolution, with the discriminator chosen slightly more than the generator on an average. This is consistent with the standard suggestion for WGAN-GP that the discriminator be given larger updates than the generator and implies that there is an overall balance that is maintained. However, the dynamic nature of the updates that is visible within intervals of lower resolution emphasize that the delicate balance between the agents is dependent on the state of the optimization. An agent's optimal response to its adversary's current strategy might require multiple optimization steps. Thus, the dynamic schedule parameterized by the controller and learned using perturbed DG, is a better candidate over a fixed schedule to enforce faster convergence for GANs.

### Implementation Details

The controller network consists of two dense layers with 128 and 64 nodes respectively, each using a tanh activation, followed by the output projection layer having two nodes with softmax activation. We use an Adam optimizer with an initial learning rate of 1e-3 to update the paramaters of the controller. We specify an episode to consist of $K$ = 30000 iterations, every 500 of which we compute perturbed DG and maintain the exponential moving average to be used in the reward computation. The value of the reward constant $\alpha$ is set to 5 and $\epsilon$ is set to 1e-5 for numerical stability. As for the task model (WGAN-GP) on 2D synthetic datasets, we use 3 dense layers with 128 nodes each having relu activation, followed by the projection layer, for both the generator and discriminator. To test for MNIST, we follow the same architecture of WGAN-GP discussed under the Image Datasets section. The dimension of the latent space is set to 100 for all tasks. The parameters of the task model are updated using an Adam optimizer with an initial learning rate of 5e-4. To compute perturbed DG, the auxilary models are optimized for 300 iterations, with the radius of the perturbation ball set to 0.25.

## Ablation Studies

### Trend of Perturbed DG across Training Iterations of GAN

We monitor the trend of perturbed DG (blue) across training iterations w.r.t to radii of perturbation ball $\sigma$ for convergence, mode collapse, and divergence settings. For each experiment, we train a classic GAN for 10000 iterations, and the average plots across 10 trials are shown in figure 8. As we move from top to bottom, $\sigma$ increases. For the convergence setting (first column), perturbed DG values increase gradually with $\sigma$. However, in mode collapse and divergence setting (second and third column), perturbed DG increases abruptly and then saturates. This result is consistent with figure 5(a) of the main paper, where we show that perturbed DG increases gradually with $\sigma$ for convergence setting but saturates immediately after a sharp increase in case of mode collapse and divergence settings. However, there we had analyzed the effect of $\sigma$ at the end of the training process; here, analysis is done during training of GAN.

### Trend of $M_1$and$M_2$ across Training Iterations of Auxiliary GAN

We study the effect of the number of iterations involved in estimating $M_1$ and $M_2$ using our approach. We train a classic GAN for 10000 iterations across the three different settings and present the variation in the final $M_1$ and $M_2$ estimates w.r.t increase in the number of iterations used in the optimization of the auxiliary models. In figure 9(a), we observe that there is no significant change to $M_1$ post 200 iterations for mode collapse and divergence settings. However, $M_1$ gradually increases and saturates for the convergence setting. A possible reason for this is that the discriminator is initially confused during convergence and optimizing it, keeping the generator fixed, enables it to learn a more discriminative decision boundary. However, the slope being marginal reflects that it is within the influence of the Nash point despite the perturbation.

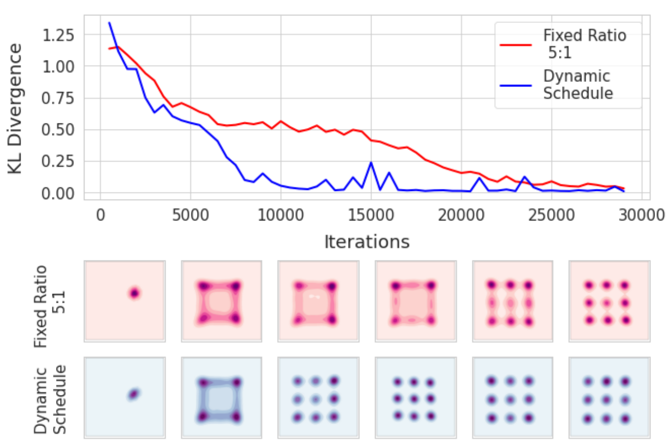
In figure 9(b) we observe that there is no significant change to $M_2$ post 300 iterations for all the three settings. The $M_1$ and $M_2$ values being close during convergence further verifies that the perturbation does not displace the agents from the influence of the Nash point.

## References

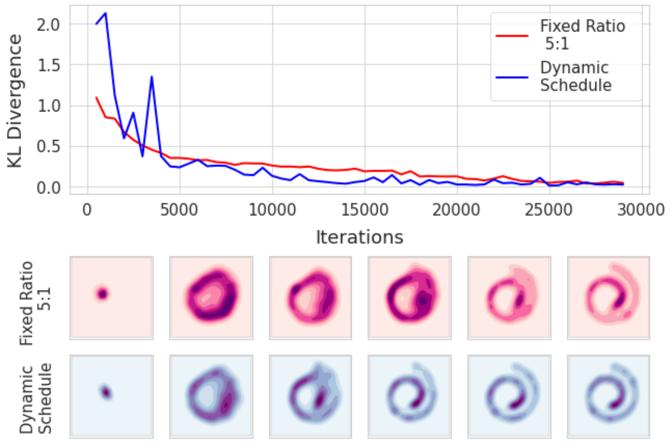
Fedus, W.; Rosca, M.; Lakshminarayanan, B.; Dai, A. M.; Mohamed, S.; and Goodfellow, I. J. 2018. Many Paths to Equilibrium: GANs Do Not Need to Decrease a Divergence At Every Step. In *ICLR*.

Williams, R. J. 1992. Simple Statistical Gradient-Following Algorithms for Connectionist Reinforcement Learning .

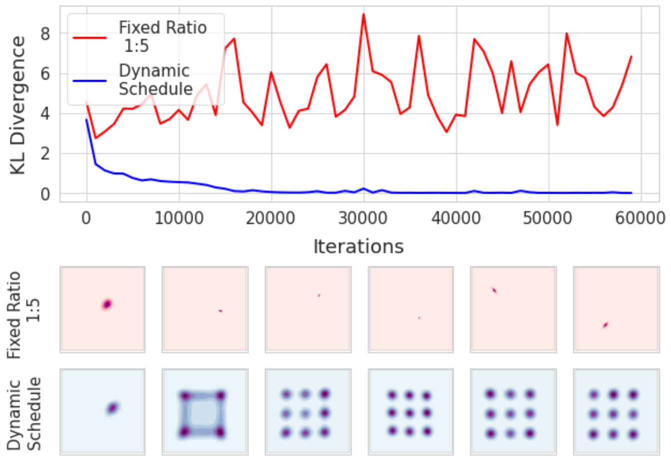
Xu, H.; Zhang, H.; Hu, Z.; Liang, X.; Salakhutdinov, R.; and Xing, E. P. 2019. AutoLoss: Learning Discrete Schedule for Alternate Optimization. In *ICLR*.



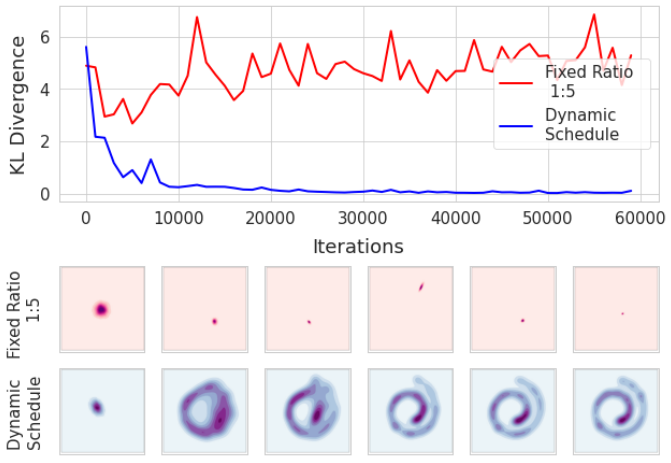
(a) GRID Dataset -Fixed Ratio 5:1



(b) SPIRAL Dataset - Fixed Ratio 5:1



(c) GRID Dataset -Fixed Ratio 1:5



(d) SPIRAL Dataset -Fixed Ratio 1:5

Figure 6: Dynamic scheduling using perturbed DG leads to faster and better convergence

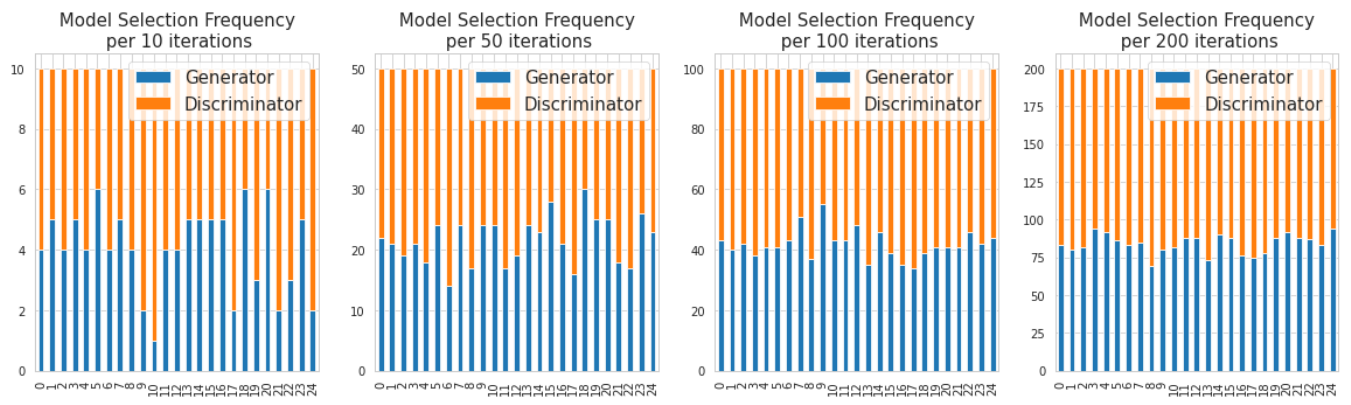


Figure 7: Dynamic Schedule : Selection Frequency of Generator and Discriminator within intervals of varying resolution. Across the x-axis are successive intervals of the specified resolution. The Y axis depicts the proportion of times each model is selected within the specified interval.

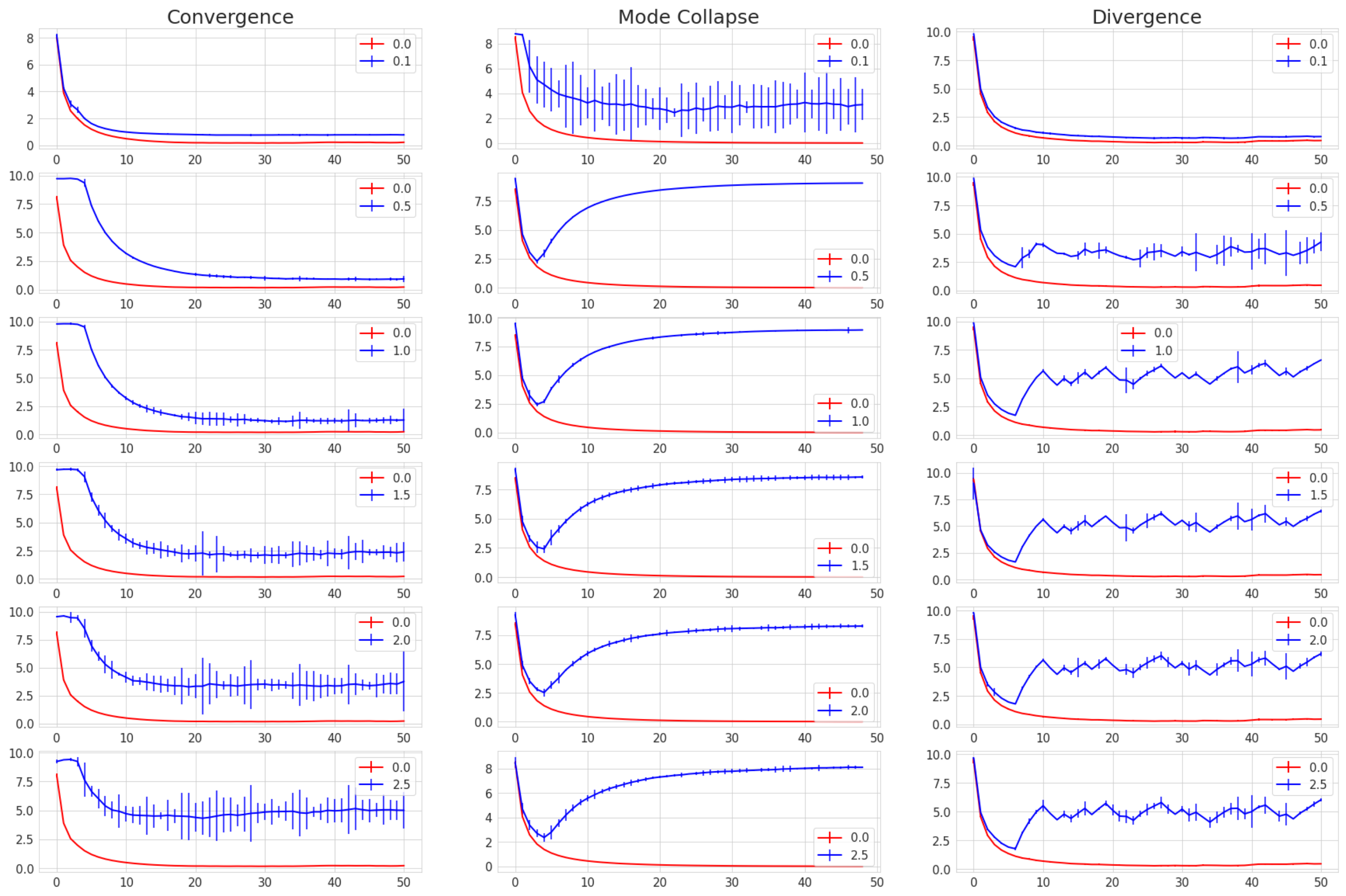
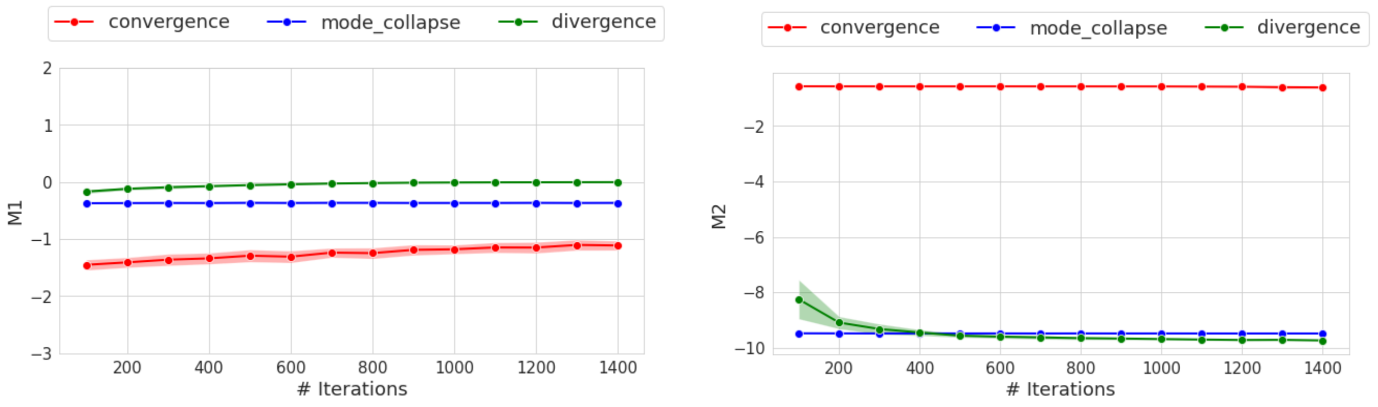


Figure 8: Trend of perturbed (blue) and vanilla (Red) DG across GAN training iterations. Rows correspond to different $\sigma$ values and columns correspond to settings. Legends in each subplot represent the radius of the perturbation ball corresponding to the curves. Curves corresponding to perturbation ball of radius zero is equivalent to the vanilla DG estimate. Within each subplot, the y-axis corresponds to the estimated duality gap and x-axis corresponds to the training step. The x-axis scale is in 200 iterations.



(a) $M_1$ Vs Iterations

(b) $M_2$ Vs Iterations

Figure 9: Trend of $M_1$ and $M_2$ w.r.t training iterations of an auxiliary GAN for convergence, mode collapse and divergence settings.