Multi-objective training of Generative Adversarial Networks with multiple discriminators

Isabela Albuquerque*, João Monteiro*, Thang Doan, Breandan Considine, Tiago Falk, and Ioannis Mitliagkas

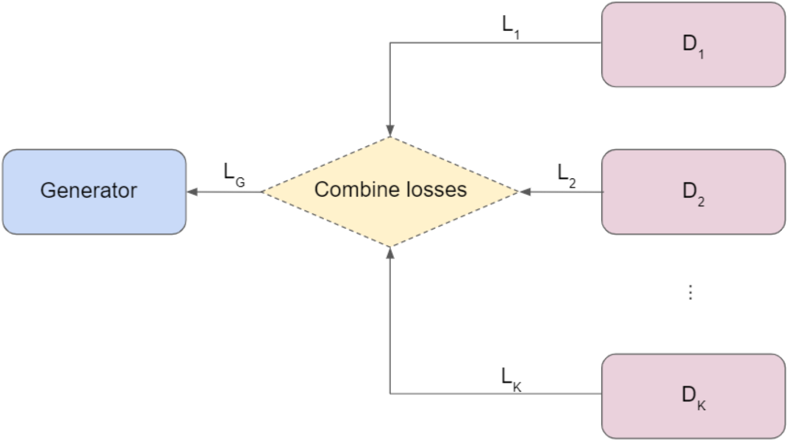
*Equal contribution

The multiple discriminators GAN setting

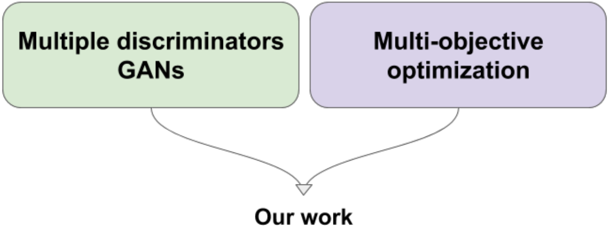
- Recent literature proposed to tackle GANs training instability* issues with multiple discriminators (Ds)
    1. Generative multi-adversarial networks, Durugkar et al. (2016)
    2. Stabilizing GANs training with multiple random projections, Neyshabur et al. (2017)
    3. Online Adaptative Curriculum Learning for GANs, Doan et al. (2018)
    4. Domain Partitioning Network, Csaba et al. (2019)

  *Mode-collapse or vanishing gradients

The multiple discriminators GAN setting

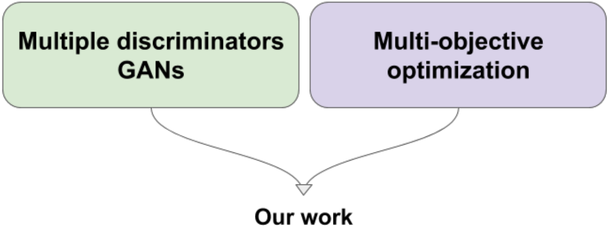


Our work



Multiple discriminators GANs
Multi-objective optimization
Our work

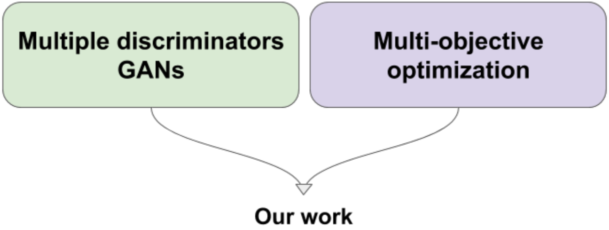
Our work



$$\min \boldsymbol{\mathcal{L}}_G(\mathbf{z}) = [l_1(\mathbf{z}), l_2(\mathbf{z}), ..., l_K(\mathbf{z})]^T$$

- Each $l_k = -\mathbb{E}_{z\sim p_z} \log D_k(G(z))$ is the loss provided by the k-th discriminator

Our work



$$\min \boldsymbol{\mathcal{L}}_G(\mathbf{z}) = [l_1(\mathbf{z}), l_2(\mathbf{z}), ..., l_K(\mathbf{z})]^T$$

- Multiple gradient descent (MGD) is a natural choice to solve this problem
  - But it might be too costly
- Alternative: maximize the hypervolume (HV) of a single solution

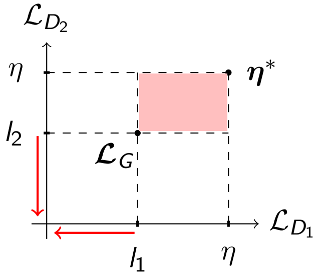
Multiple gradient descent

- Seeks a Pareto-stationary solution
- Two steps:
  1. Find a common descent direction $\forall l_k$
     1.1 Minimum norm element within the convex hull of all $\nabla l_k(\mathbf{x})$
  2. Update the parameters with $\mathbf{x}_{t+1} = \mathbf{x}_t - \lambda \frac{\mathbf{w}_t^*}{||\mathbf{w}_t^*||}$, where

$$\mathbf{w}_t^* = \text{argmin}||\mathbf{w}||^2, \quad \mathbf{w} = \sum_{k=1}^{K} \alpha_k \nabla l_k(\mathbf{x}_t),$$

$$\text{s.t.} \quad \sum_{k=1}^{K} \alpha_k = 1, \quad \alpha_k \geq 0 \quad \forall k$$

Hypervolume maximization for training GANs

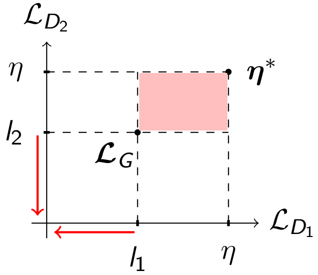


Hypervolume maximization for training GANs

$$\mathcal{L}_G = -\log\left(\prod_{k=1}^{K}(\eta - l_k)\right)$$

$$\mathcal{L}_G = -\sum_{k=1}^{K}\log(\eta - l_k)$$

$$\frac{\partial \mathcal{L}_G}{\partial \theta} = \sum_{k=1}^{K}\boxed{\frac{1}{\eta - l_k}}\frac{\partial l_k}{\partial \theta}$$

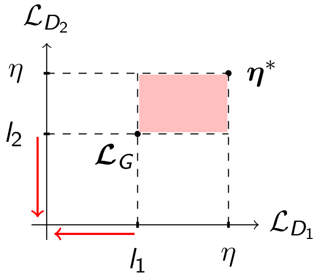


Hypervolume maximization for training GANs

$$\mathcal{L}_G = -\log\left(\prod_{k=1}^{K}(\eta - l_k)\right)$$

$$\mathcal{L}_G = -\sum_{k=1}^{K}\log(\eta - l_k)$$

$$\frac{\partial \mathcal{L}_G}{\partial \theta} = \sum_{k=1}^{K}\boxed{\frac{1}{\eta - l_k}}\frac{\partial l_k}{\partial \theta}$$



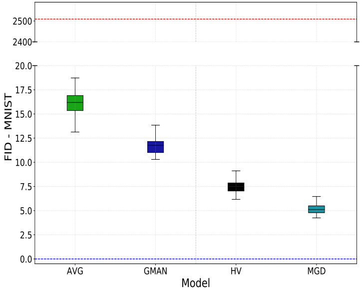
$$\eta^t = \delta \max_k \{l_k^t\}, \quad \delta > 1$$

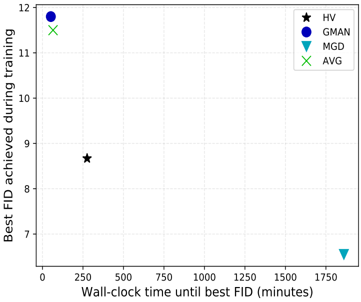
MGD vs. HV maximization vs. Average loss minimization

- MGD seeks a Pareto-stationary solution
  - $\mathbf{x}_{t+1} \prec \mathbf{x}_t$
- HV maximization seeks Pareto-optimal solutions
  - $\text{HV}(\mathbf{x}_{t+1}) > \text{HV}(\mathbf{x}_t)$
  - For the single-solution case, central regions of the Pareto-front are preferred
- Average loss minimization does not enforce equally good individual losses
  - Might be problematic in case there is a trade-off between discriminators

MNIST

- Same architecture, hyperparameters, and initialization for all methods
- 8 Ds, 100 epochs
- FID was calculated using a LeNet trained on MNIST until 98% test accuracy





Upscaled CIFAR-10 - Computational cost

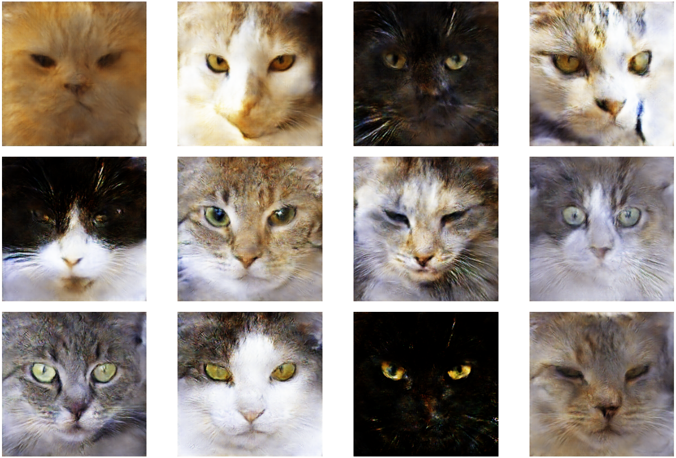
- Different GANs with both 1 and 24 Ds + HV
- Same architecture and initialization for all methods
- Comparison of minimum FID obtained during training, along with computation cost in terms of time and space

	# Disc.	FID-ResNet	FLOPS*	Memory
DCGAN	1	4.22	8e10	1292
	24	1.89	5e11	5671
LSGAN	1	4.55	8e10	1303
	24	1.91	5e11	5682
HingeGAN	1	6.17	8e10	1303
	24	2.25	5e11	5682

*Floating point operations per second

- Additional cost $\rightarrow$ performance improvement

Cats 256 × 256



Thank you!

Questions? Come to our poster! #4

Code: `https://github.com/joaomonteirof/hGAN`