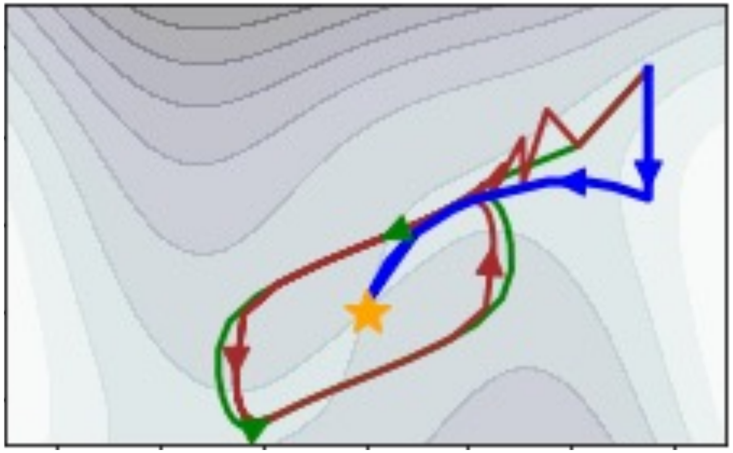
A Convergent and Dimension-Independent Min-Max Optimization Algorithm



Vijay Keswani
Yale

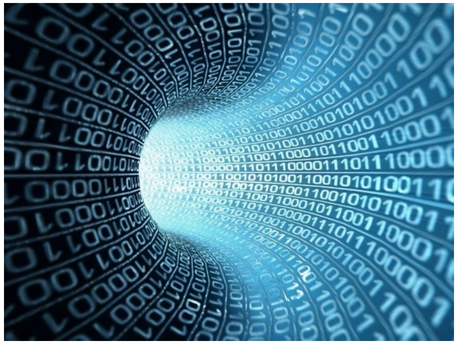
Oren Mangoubi
WPI

Sushant Sachdeva
University of Toronto

Nisheeth Vishnoi
Yale

ICML 2022

Learning and Minimization



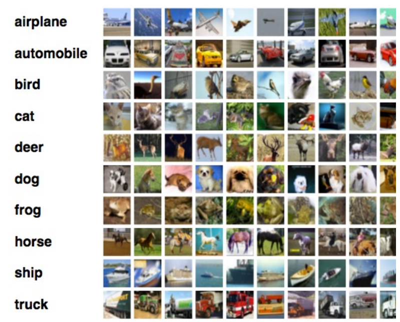
From ML to Optimization:

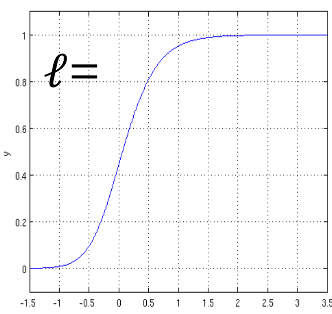
Given $f\colon \mathbb{R}^d \to \mathbb{R}$

$$\min_x f(x)$$

Availability of **large, real-world datasets** has given rise to complex, **nonconvex**, objective functions in high dimensions e.g.

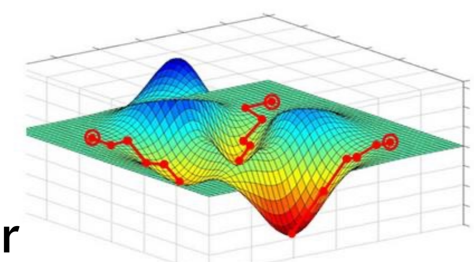
$f(x) = \sum_i \ell_i(x, D_i)$





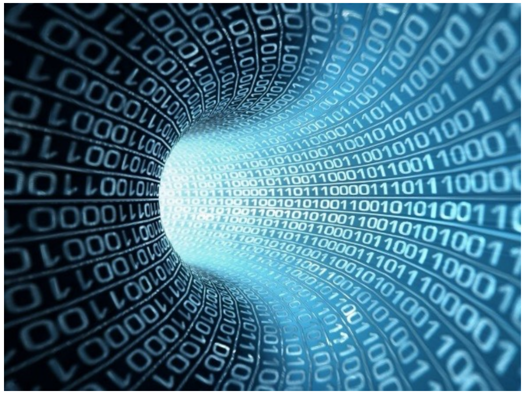
Simple examples of nonconvex f where *any* algorithm with access to oracles for f, ∇f, $\nabla^2 f$ requires exponential-in-d oracle calls to find global min

Even if f is given as a neural network, minimizing f is still hard [Rivest, Blum, '89]



Many efficient local minimization algorithms for nonconvex minimization

"Robust" Learning and "Min-Max" Optimization



Applications:

- Privacy
- Bias
- Adversarial attacks
- Unsupervised learning
- …

From Robust Learning to Min-Max Optimization: $f: \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$

$$\min_x \max_y f(x, y)$$

Learner: model parameters

Adversary: Perturbations, …

Convex-concave setting (reasonably) well-understood (starting with [von Neumann, 1928]…)

In ML applications: f is nonconvex in x and nonconcave in y

Bottleneck: (Locally) convergent algorithms for min-max models?

ε-Local Minima

- **Definition:** ε −local minimum [Nesterov, Polyak, '06]:
  $x^\star$ is a first-order (second-order) ε −local minimum of $f(x)$ if
  $$||\nabla f(x^\star)|| < \varepsilon \qquad \nabla^2 f(x^\star) \succcurlyeq -\sqrt{\varepsilon} I$$

At any point which is *not* a first-order (or second-order) ε −local minimum, can decrease f by roughly ε in $\mathrm{poly}\left(\frac{1}{\varepsilon}, L, \log(d)\right)$ gradient and/or Hessian evaluations! (*If f is L-smooth*)

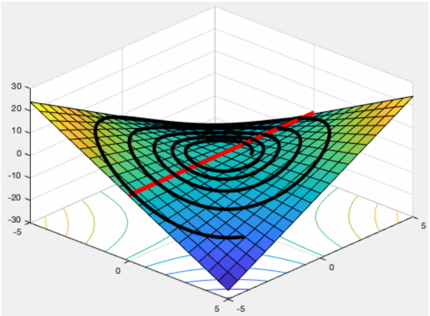
- $\mathrm{poly}\left(\frac{1}{\varepsilon}, L, \log(d)\right)$ Newton's method with cubic regularization (need Hessian-vector product) [Nesterov, Polyak, '06]
- Stochastic gradient descent (only gradient evaluations)
  - $\mathrm{poly}\left(\frac{1}{\varepsilon}, L, d\right)$ [Ge, Huang, Jin Yuan, '15]
  - $\mathrm{poly}\left(\frac{1}{\varepsilon}, L, \log(d)\right)$ [Jin, Ge, Netrapalli, Kakade, Jordan, '17]

Local Equilibria for Min-Max

Many prior attempts, e.g., ε-local min-max: a point $(x^\star, y^\star)$ where

1) $y^\star$ is a ε-local maximum for $f(x^\star,\cdot)$ and
2) $x^\star$ is a ε-local minimum for $f(\cdot, y^\star)$

- Simple examples where such points don't exist [see Jin, Netrapalli, Jordan '19] and hard to find even when they exist [see Daskalakis, Skoulakis, Zampetakis '21]
- *Leads to convergence problems in algorithms such as gradient descent-ascent (GDA), opt. mirror desc.*



Convergence for "local" min-max algorithms require strong assumptions, e.g.:

- GDA [Heusel, Ramsauer, Unterthiner, Nessler, Hochreiter '17] (special starting point)
- Optimistic mirror descent [Daskalakis, Panageas '18] (f bilinear, or "coherence")
  Hamiltonian descent [Abernathy, Lai, Wibisono '19] (f to be sufficiently bilinear)
- Other Algorithms [Thekumparampil, Jain, Netrapali, Oh '19], [Rafique, Liu, Lin, Yang '18] (concave in y)

Computationally restricted equilibrium: *min-max equilibrium for agents computationally restricted to 2nd-order algorithms [Mangoubi, Vishnoi, '21]*

- *Algorithm converges for **any** smooth/bounded f, from any initial point*
- *Runtime bound is polynomial-in-d, requires access to Hessian $\nabla^2 f$*

This Paper

Key Idea : Place ***first-order*** computational restrictions on max-agent (adversary)

Definition: $(\varepsilon, \delta, \omega, Q)$-min-max equilibrium under first-order max-agent *(coming up)*

Theorem: Given access to $f: \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$, its (stochastic) gradient, and a sampling oracle for a proposal distribution Q. Suppose f is L-smooth and uniformly bounded by $b>0$. Then given any initial point, our algorithm returns an $(\varepsilon, \delta, \omega, Q)$- equilibrium $(x^\star, y^\star)$ of f in a number of function, gradient, and sampling oracle evaluations that is *poly*$\left(L, b, \frac{1}{\varepsilon}, \frac{1}{\delta}, \frac{1}{\omega}\right)$ and does not depend on the dimension d.

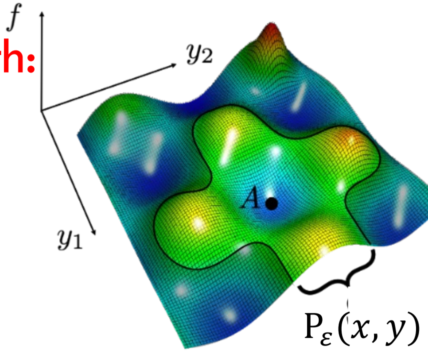
- No additional assumptions on starting point, concavity, coherence…
- Equilibrium exists for every bounded and smooth f
- Number of gradient evaluations **does not depend on dimension d**

ε-increasing Paths and ε-Equilibria

Starting at (x, y), update y to w using a (first-order) ε-increasing path:
Any unit speed path $\gamma: [0, \tau] \to \mathbb{R}^d$ *s.t.*

$$\frac{d}{dt} f(x, \gamma(t)) \geq \varepsilon$$

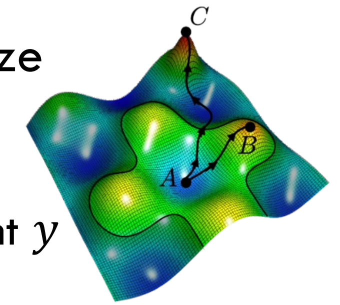
ε-increasing paths model classes of 1st-order optimization algorithms!



If adversary is restricted to ε-increasing paths, min-agent seeks to minimize

$$\mathcal{L}(x, y) := \max_{z \in P_\varepsilon(x,y)} f(x, z),$$

Where $P_\varepsilon(x, y)$ is set of points reachable by ε-increasing path from an initial point y



A first attempt to define first-order equilibrium for this framework:

- $\|\nabla_y f(x^\star, y^\star)\| \leq \varepsilon$ and $\|\nabla_x \mathcal{L}(x^\star, y^\star)\| \leq \varepsilon$

But $\mathcal{L}$ may be discontinuous!

$(\varepsilon, \delta, \omega, Q)$-min-max equilibrium :

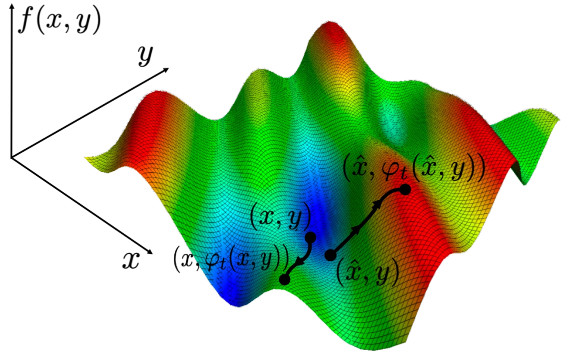
- $\mathbb{P}_{\Delta \sim Q_{x^\star, y^\star}}(\mathcal{L}(x^\star + \Delta, y^\star) < \mathcal{L}(x^\star, y^\star) - \delta) \leq \omega$
- $y^\star \in \operatorname{argmax}_{y \in P_\varepsilon(x^\star, y^\star)} f(x^\star, y)$

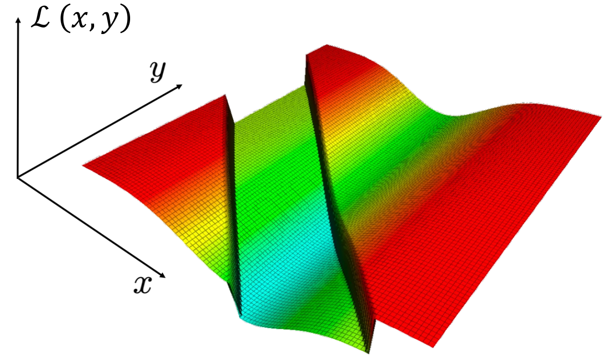
$Q_{x,y}$ is a proposal distribution used by the min-player to search for updates

How to choose Q to minimize discontinuous $\mathcal{L}$?

First-order method for Minimizing Discontinuous $\mathcal{L}$

(Common) problem: $\mathcal{L}\,(x,y)$ may be **discontinuous** in x





Problem: Even where $\mathcal{L}$ is differentiable, don't have access to its gradient $\nabla_x \mathcal{L}$

How can min-agent minimize $\mathcal{L}$ to update $(\hat{x},\hat{y})$ via **first-order** algorithm?

Solution:

- Min-agent proposes random updates $\hat{x} + \Delta$ from a distribution $\Delta \sim Q$
- Roughly speaking, if $\mathcal{L}(\hat{x}+\Delta, y) < \mathcal{L}(x,y)$, accept the update. Otherwise, propose a new random update.

In practice, we observe that choosing Q to be distribution of stochastic gradients $-\nabla_x f$ leads to equilibria with good learning outcomes

Algorithm

Input: Initial point $(\hat{x}, \hat{y})$, $f: \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$

For $i = 0,1,2, \ldots$

1. Sample $\Delta \sim Q_{\hat{x},\hat{y}}$

   In practice we choose Q to be distribution of stochastic (batch) gradients for $-\nabla_x f$

2. Propose min-player update: $x \leftarrow \hat{x} + \Delta$

3. Compute max-agent's response, y, by running gradient ascent on $f(x,\cdot)$, starting at $\hat{y}$, until a point y is reached s.t. $||\nabla_y f(x,y)|| < \varepsilon$

4. If $\mathcal{L}(x,y) < \mathcal{L}(\hat{x},\hat{y}) - \delta$, accept proposed update $(\hat{x},\hat{y}) \leftarrow (x,y)$

5. If no "accept" in previous $\frac{1}{\omega}$ iterations of for loop, return $(\hat{x},\hat{y})$ and halt

---

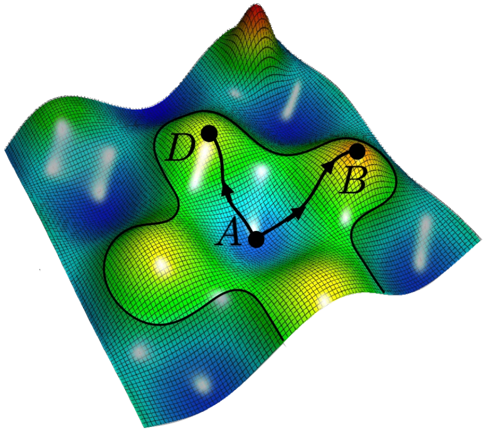
Runtime: Roughly, $\mathcal{L}(\hat{x},\hat{y})$ decreases by at least δ each time proposal is accepted, which occurs at least every $\frac{1}{\omega}$ Iterations. Since f (and hence $\mathcal{L}$) is b-bounded, algorithm terminates after $\leq \frac{b}{\delta\omega}$ iterations.

At each iteration, Q is sampled once, and gradient ascent computes $\mathrm{poly}(L, b, 1/\varepsilon)$ gradients. Thus, total runtime is $\mathrm{poly}(L, b, 1/\varepsilon, 1/\omega, 1/\delta)$ gradient/sampling oracle calls.

Equilibrium: The point $(x^\star, y^\star)$ reached by the algorithm is a first-order ε-local max for $f(x^\star,\cdot)$, and satisfies $\mathbb{P}_{\Delta \sim Q_{x^\star,y^\star}}(\mathcal{L}(x^\star + \Delta, y^\star) < \mathcal{L}(x^\star, y^\star) - \delta) \leq \omega$.

Convergence to Equilibrium

Problem: $\mathcal{L}(x, y)$ may not be tractable to compute at all (x, y)



- Adversary can choose to use *any* ε-increasing path
- Finding the max over all these paths is intractable

Solution: Have the min-agent minimize a lower bound $h(x, y) \leq \mathcal{L}(x, y)$, obtained with just one ε-increasing path

We show that, at any points $(x, y^\star)$ where $y^\star$ is ε-stationary point of $f(x, \cdot)$,

1. $h(x, y^\star) = \mathcal{L}(x, y^\star) = f(x, y^\star)$
   (because any ε-increasing path initialized at ε-stationary point $y^\star$ remains at $y^\star$)

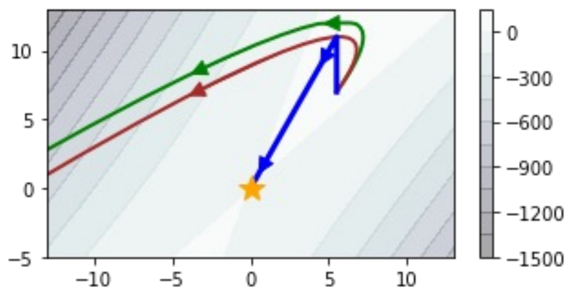
2. If $h(x^\star + \Delta, y^\star) > h(x^\star, y^\star) - \delta$ then $\mathcal{L}(x^\star + \Delta, y^\star) > \mathcal{L}(x^\star, y^\star) - \delta$
   Because $\mathcal{L}(x^\star + \Delta, y^\star) \geq h(x^\star + \Delta, y^\star)$ *(since $h \leq g$)*
   $> h(x^\star, y^\star) - \delta$
   $= f(x^\star, y^\star) - \delta$ *(by (1))*

Empirical Results: 2-D functions and synthetic data

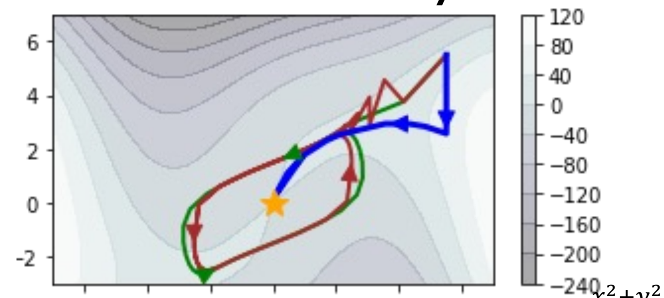
2-dimensional min-max objectives bounded above in y

Global min−max at $(0,0)$**:**

- GDA and OMD cycle or diverge to ∞
- Our algorithm converged to global min-max $(0,0)$



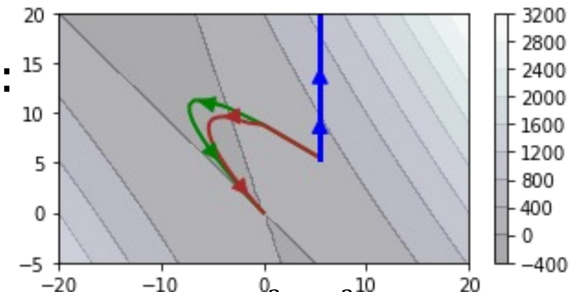
$F_1(x,y) = -3x^2 - y^2 + 4xy$



$F_3(x,y) = (4x^2 - (y - 3x + .05x^3) - .1y^4)e^{-\frac{x^2+y^2}{100}}$

Global min-max (value) at $+\infty$**:**

- GDA and OMD go to point which is not global min-max
- Our algorithm goes to $+\infty$



$F_3(x,y) = 3x^2 + y^2 + 4xy$

GDA

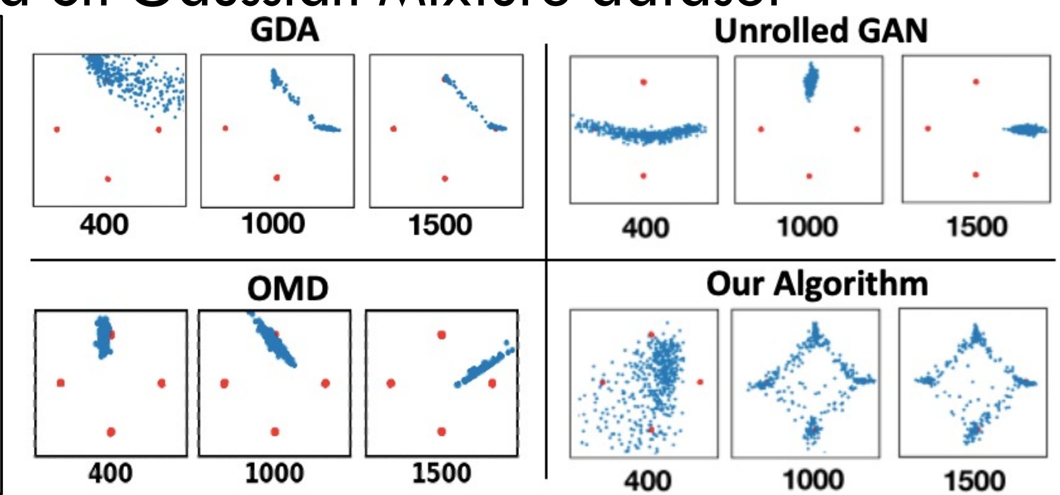
OMD [Daskalakis, Ilyas, Syrgkanis, Zeng, '18]

Our Algorithm

GANs trained on Gaussian Mixture dataset

	Number of modes learnt			
Method	1	2	3	4
Our Algorithm	0	0.15	0.15	**0.70**
GDA ($k = 1$)	0.95	0.05	0	0
GDA ($k = 6$)	0.05	0.75	0	0.20
OMD	0.80	0.20	0	0
Unrolled-GAN	0.75	0.15	0.10	0

[Metz, Poole, Pfau, Sohl-Dickstein, '17]



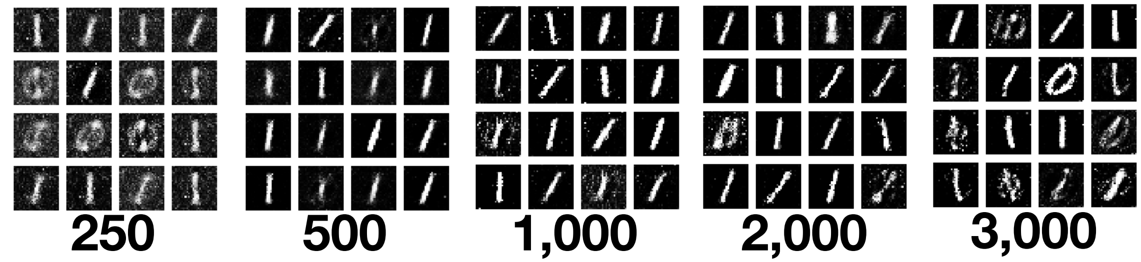
Empirical Results: Real-world datasets

GANs trained on 01-MNIST dataset

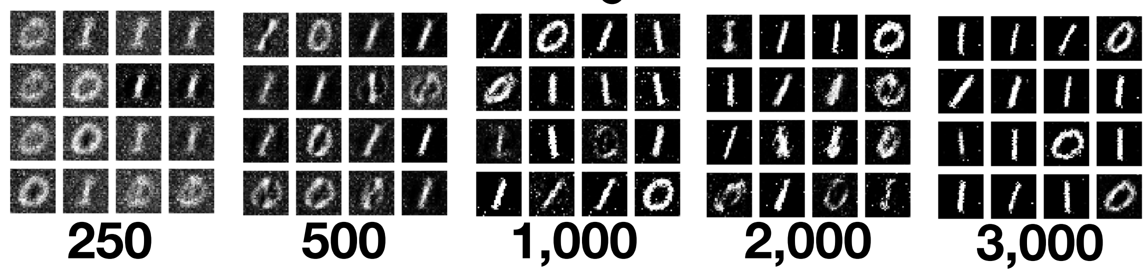
Mode collapse: by the 1000th iteration,

- Our algorithm generated both digits in all the training runs
- GDA did so in **22%** of the runs

GDA



Our Algorithm

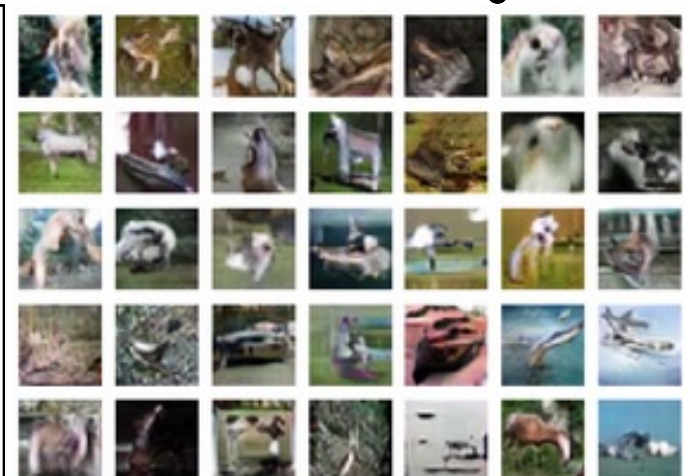


GANs trained on CIFAR-10 dataset

Mean Inception score (standard deviation)

Method	Iteration 5000	25000	50000
Our Algorithm	2.71 (0.28)	4.10 (0.35)	**4.68** (0.39)
GDA	2.80 (0.52)	4.28 (0.77)	4.51 (0.86)
OMD	1.60 (0.18)	1.73 (0.25)	1.96 (0.26)

Images generated by GAN trained with our algorithm



Conclusions

- New **first-order** computationally feasible alternative to min-max optimization

- **Key idea:** constrain max-agent to ε-increasing paths, which model first-order optimization algorithms

- **Dimension-independent bounds:** algorithm finds ε-equilibrium point in $\mathrm{poly}\left(\frac{1}{\varepsilon}, L\right)$ gradient/sampling oracle evaluations
  - only assume that f is L-smooth and bounded
  - No additional assumptions on starting point, concavity, coherence, etc.
  - In practice, update $\Delta \sim Q$ can be computed as stochastic gradient for $-\nabla_x f$

- **Previous work:** Use paths which model **second-order** greedy algorithms, converges to a **second-order** $\varepsilon -$ equilibrium in $\mathrm{poly}\left(d, \frac{1}{\varepsilon}, L\right)$ gradient and Hessian evaluations [Mangoubi, Vishnoi, '21]

- **Open problem:** Can a second-order equilibrium be found in $\mathrm{polylog}(d)$ gradient evaluations (and without access to Hessian)?

Thanks!