New (Optimization) Perspectives on GANs

Gauthier Gidel

I. A Variational Inequality Perspective on GANs.

II. Reducing Noise in GANs with Variance Reduced Methods.

A Variational Inequality Perspective on GANs

Gauthier Gidel[*1], Hugo Berard[*12], Gaëtan Vignoud[1],
Pascal Vincent[12], Simon Lacoste-Julien[1]

**equal contribution*

[1] Mila, Université de Montréal

[2] Facebook AI Research (FAIR), Montréal

facebook Artificial Intelligence Research

Université de Montréal

Mila

Hugo Berard

Gaëtan Vignoud

Pascal Vincent

Simon Lacoste-Julien

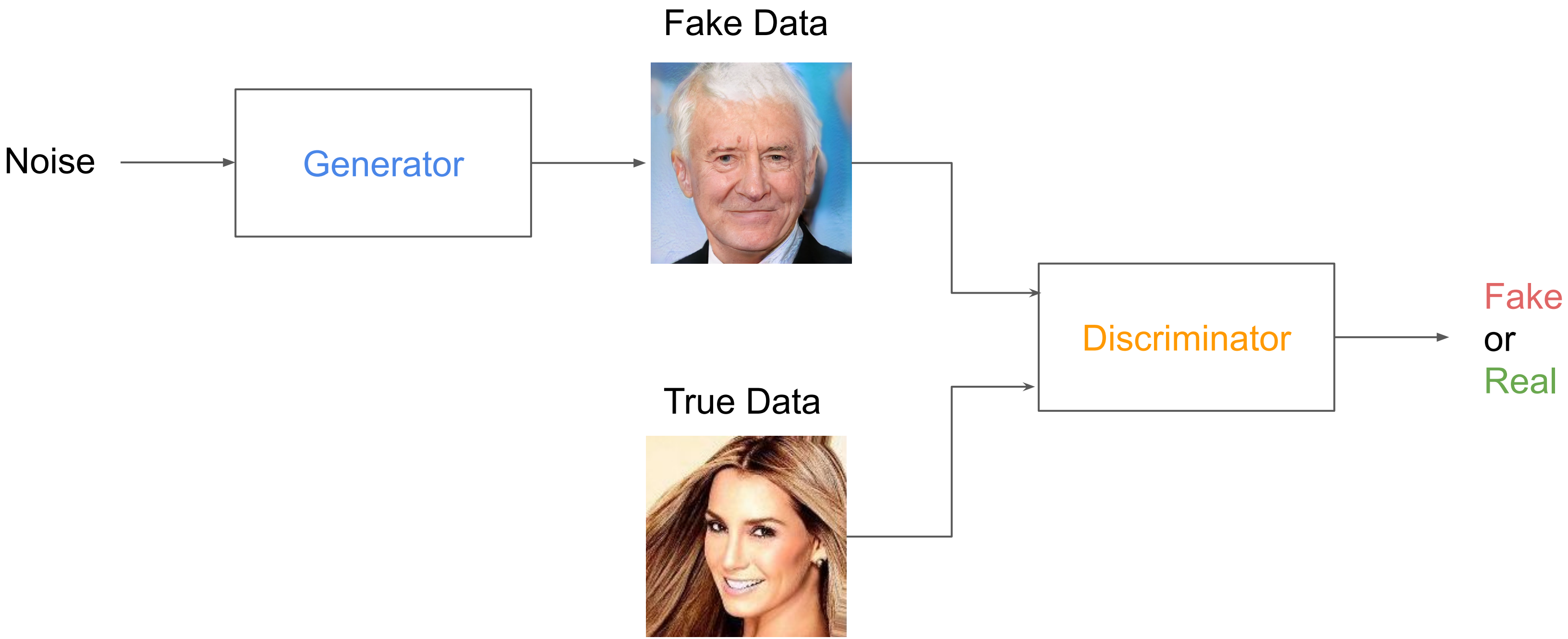
1. Quick Recap on GANs and two-player games.
2. GAN as a Variational Inequality Problem.
3. Optimization of Variational Inequality.
4. Experimental results.
5. Conclusion.

NB: All the citations in this talk are in my arXiv submission.

Quick recap on Generative Adversarial Networks (GANs) (and two-player games)

Generative Adversarial Networks (GANs)

[Goodfelow et al. NIPS 2014]



Generative Adversarial Networks (GANs)
[Goodfelow et al. NIPS 2014]

Discriminator

Generator

$$\min_{\theta} \underbrace{\max_{\phi} \mathbb{E}_{x \sim p_{\mathcal{D}}}[\log(D_{\phi}(x))] + \mathbb{E}_{z \sim p_{\mathcal{Z}}}[\log(1 - D_{\phi}(G_{\theta}(z)))]}$$

If D is non-parametric: $L(\theta) = \mathrm{JSD}(p_{\mathcal{D}} || p_{\theta})$

Non-saturating GAN: “much stronger gradient in early learning”

Loss of Generator

$$\overbrace{\min_{\theta} -\mathbb{E}_{z \sim p_{\mathcal{Z}}}[\log(D_{\phi}(G_{\theta}(z)))]}$$

Loss of Discriminator

$$\overbrace{\max_{\phi} \mathbb{E}_{x \sim p_{\mathcal{D}}}[\log(D_{\phi}(x))] + \mathbb{E}_{z \sim p_{\mathcal{Z}}}[\log(1 - D_{\phi}(G_{\theta}(z)))]}$$

Two-player Games

Player 1 Player 2

$$\boldsymbol{\theta}^* \in \arg\min_{\boldsymbol{\theta}\in\Theta} \mathcal{L}^{(\boldsymbol{\theta})}(\boldsymbol{\theta}, \boldsymbol{\varphi}^*) \quad \text{and} \quad \boldsymbol{\varphi}^* \in \arg\min_{\boldsymbol{\varphi}\in\Phi} \mathcal{L}^{(\boldsymbol{\varphi})}(\boldsymbol{\theta}^*, \boldsymbol{\varphi})$$

Zero-sum game if: $\mathcal{L}^{(\boldsymbol{\theta})} = -\mathcal{L}^{(\boldsymbol{\varphi})}$ also called *Saddle Point* (SP).

<u>Example</u>: WGAN formulation [Arjovsky et al. 2017]

$$\min_{\theta} \max_{\phi, ||f_\phi||_L \leq 1} \underbrace{\mathbb{E}_{x\sim p_{\mathcal{D}}}[f_\phi(x)] - \mathbb{E}_{z\sim p_{\mathcal{Z}}}[f_\phi(g_\theta(z)))]}_{\mathcal{L}^{(\boldsymbol{\theta})} = -\mathcal{L}^{(\boldsymbol{\varphi})}}$$

Two-player Games

Player 1 Player 2

$$\theta^* \in \arg\min_{\theta \in \Theta} \mathcal{L}^{(\theta)}(\theta, \varphi^*) \quad \text{and} \quad \varphi^* \in \arg\min_{\varphi \in \Phi} \mathcal{L}^{(\varphi)}(\theta^*, \varphi)$$

- In games we want to **converge** to the Saddle Point.
- Different from **single** objective **minimization** where we want to avoid saddle points.
- ~~Saddle point~~ -> **Zero-sum game (or Minmax)**

Two-player Games

Player 1 Player 2

$$\boldsymbol{\theta}^* \in \arg\min_{\boldsymbol{\theta}\in\Theta} \mathcal{L}^{(\boldsymbol{\theta})}(\boldsymbol{\theta}, \boldsymbol{\varphi}^*) \quad \text{and} \quad \boldsymbol{\varphi}^* \in \arg\min_{\boldsymbol{\varphi}\in\Phi} \mathcal{L}^{(\boldsymbol{\varphi})}(\boldsymbol{\theta}^*, \boldsymbol{\varphi})$$

Non zero-sum game if we **do not** have: $\mathcal{L}^{(\boldsymbol{\theta})} = -\mathcal{L}^{(\boldsymbol{\varphi})}$

Example: Non-saturating GAN: [Goodfellow et al. 2014]

Loss of Generator Loss of Discriminator

$$\min_{\theta} -\mathbb{E}_{z\sim p_{\mathcal{Z}}}[\log(D_\phi(G_\theta(z)))] \qquad \max_{\phi} \mathbb{E}_{x\sim p_{\mathcal{D}}}[\log(D_\phi(x))] + \mathbb{E}_{z\sim p_{\mathcal{Z}}}[\log(1 - D_\phi(G_\theta(z)))]$$

Minmax training is ~~hard~~ different !

Minmax training is ~~hard~~ different !

(You can replace “minmax” with two-player games)

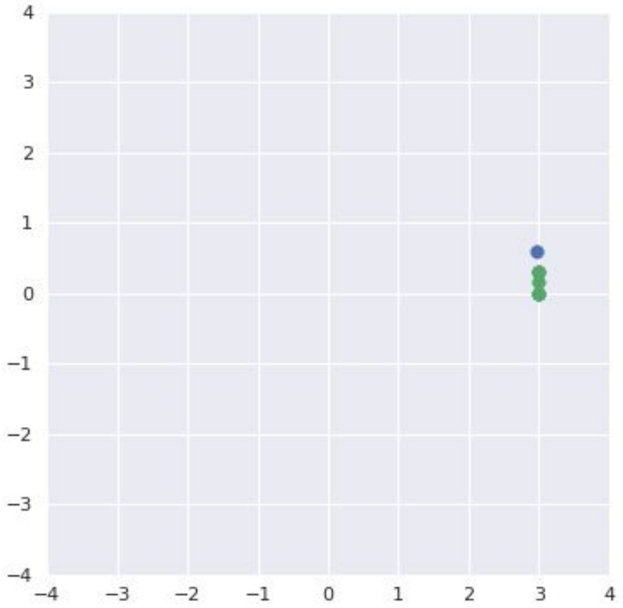
"Minmax Training is Hard …"

Example: WGAN with **linear** discriminator and generator

Gradient vector field: $F(\theta, \phi) = \begin{pmatrix} \nabla_\theta L_\theta(\theta, \phi) \\ \nabla_\phi L_\phi(\theta, \phi) \end{pmatrix}$

Bilinear saddle point = Linear in θ and ϕ

⇒ "Cycling behavior" (see right).



$$\min_{\theta} \max_{\phi, ||f_\phi||_L \leq 1} \phi^T \mathbb{E}_{x \sim p_{\mathcal{D}}}[x] - \phi^T \theta \mathbb{E}_{z \sim p_{\mathcal{Z}}}[z]$$

Generative Adversarial Networks as a Variational Inequality Problem (VIP)

GANs as a Variational Inequality

New perspective for GANs:

- Based on **stationary conditions.**
- Relates to vast literature with standard algorithms.

Nash-Equilibrium: $\begin{cases} \theta^* = \arg\min_\theta L_\theta(\theta, \phi^*) \\ \phi^* = \arg\min_\phi L_\phi(\theta^*, \phi) \end{cases}$ ← No player can improve its cost

Stationary Conditions: $\begin{cases} \nabla_\theta L_\theta(\theta^*, \phi^*)^T(\theta - \theta^*) \geq 0 \\ \nabla_\phi L_\phi(\theta^*, \phi^*)^T(\phi - \phi^*) \geq 0 \end{cases}$ $\quad \forall (\theta, \phi) \in \Theta \times \Phi$

can be **constraint sets**.

GANs as a Variational Inequality

Nash-Equilibrium:

$$\begin{cases} \theta^* = \arg\min_\theta L_\theta(\theta, \phi^*) \\ \phi^* = \arg\min_\phi L_\phi(\theta^*, \phi) \end{cases}$$

$\Longleftrightarrow$

Stationary Conditions:

$$\begin{cases} \nabla_\theta L_\theta(\theta^*, \phi^*)^T(\theta - \theta^*) \geq 0 \\ \nabla_\phi L_\phi(\theta^*, \phi^*)^T(\phi - \phi^*) \geq 0 \end{cases} \quad \forall (\theta, \phi) \in \Theta \times \Phi$$

Same problem but **different** perspective.

Joint Minimization vs. **Stationary point**

GANs as a Variational Inequality

Stationary Conditions: $\begin{cases} \nabla_\theta L_\theta(\theta^*, \phi^*)^T(\theta - \theta^*) \geq 0 \\ \nabla_\phi L_\phi(\theta^*, \phi^*)^T(\phi - \phi^*) \geq 0 \end{cases} \quad \forall (\theta, \phi) \in \Theta \times \Phi$

Can be written as: $F(\omega) = \begin{pmatrix} \nabla_\theta L_\theta(\omega) \\ \nabla_\phi L_\phi(\omega) \end{pmatrix}$

$\omega = (\theta, \phi)$

$$F(\omega^*)^T(\omega - \omega^*) \geq 0 \quad \forall \omega \in \Omega$$

ω^* solves the **Variational Inequality**

GANs as a Variational Inequality

Stationary Conditions: $$F(\omega^*)^T(\omega - \omega^*) \geq 0 \quad \forall \omega \in \Omega$$

Unconstrained (or optimum in the interior):

$$\|\nabla_{\boldsymbol{\theta}} \mathcal{L}^{(\boldsymbol{\theta})}(\boldsymbol{\theta}^*, \boldsymbol{\varphi}^*)\| = \|\nabla_{\boldsymbol{\varphi}} \mathcal{L}^{(\boldsymbol{\varphi})}(\boldsymbol{\theta}^*, \boldsymbol{\varphi}^*)\| = 0\,.$$

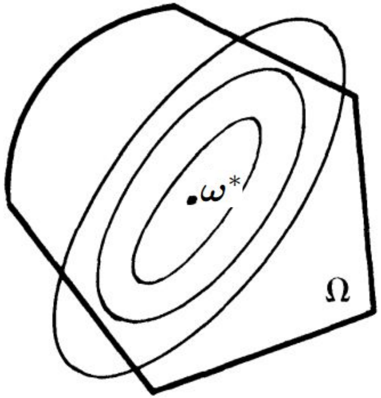


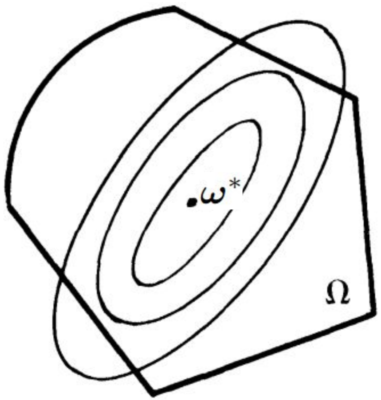
Figure from [Dunn 1979]

GANs as a Variational Inequality

Stationary Conditions: $F(\omega^*)^T(\omega - \omega^*) \geq 0 \quad \forall \omega \in \Omega$

Unconstrained (or ω* in the interior):

$$\|\nabla_{\boldsymbol{\theta}} \mathcal{L}^{(\boldsymbol{\theta})}(\boldsymbol{\theta}^*, \boldsymbol{\varphi}^*)\| = \|\nabla_{\boldsymbol{\varphi}} \mathcal{L}^{(\boldsymbol{\varphi})}(\boldsymbol{\theta}^*, \boldsymbol{\varphi}^*)\| = 0\,.$$



Constrained and ω* on the boundary:

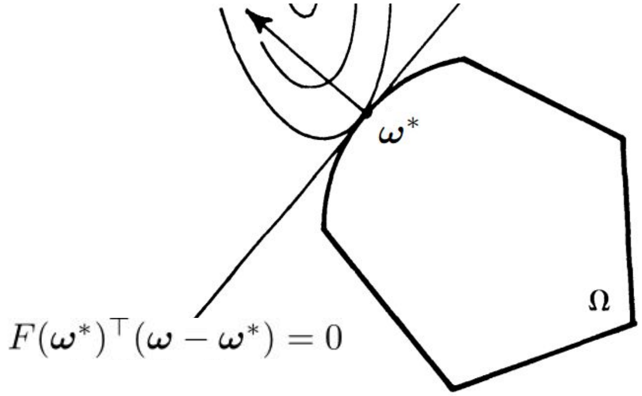


Figure from [Dunn 1979]

GANs as a Variational Inequality

Takeaways:

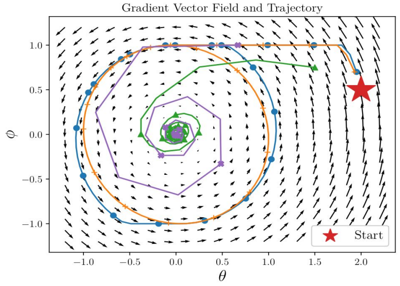
- GAN can be formulated as a **Variational Inequality.**
- Encompass <u>most</u> of GANs formulations.
- **Standard algorithms** from Variational Inequality can be used for GANs.
- **Theoretical Guarantees** (for convex and <u>*stochastic*</u> cost functions).

$$\begin{cases} \theta^* = \arg\min_{\theta} L_\theta(\theta, \phi^*) \\ \phi^* = \arg\min_{\phi} L_\phi(\theta^*, \phi) \end{cases}$$

$$\downarrow$$

$$F(\omega^*)^T(\omega - \omega^*) \geq 0 \quad \forall \omega \in \Omega$$

$$\downarrow$$



Techniques to optimize VIP
(Batch setting)

Standard Algorithms from Variational Inequality

Method 1: **Averaging**

- Converge even for *"cycling behavior"*.
- Easy to implement. (out of the training loop)
- Can be combined with any method.

$$\bar{\omega}_T \overset{\text{def}}{=} \frac{\sum_{t=0}^{T-1} \rho_t \omega_t}{S_T}, \quad S_T \overset{\text{def}}{=} \sum_{t=0}^{T-1} \rho_t \,.$$

Averaging schemes can be efficiently implemented in an **online** fashion:

$$\bar{\omega}_t = (1 - \tilde{\rho}_t)\bar{\omega}_{t-1} + \tilde{\rho}_t \omega_t \quad \text{where} \quad 0 \leq \tilde{\rho}_t \leq 1 \,.$$

Standard Algorithms from Variational Inequality

Method 1: **Averaging**

- Converge even for *"cycling behavior"*.
- Easy to implement. (out of the training loop)
- Can be combined with any method.

General Online averaging: $\bar{\omega}_t = (1-\tilde{\rho}_t)\bar{\omega}_{t-1} + \tilde{\rho}_t \omega_t \quad \text{where} \quad 0 \leq \tilde{\rho}_t \leq 1\,.$

Example 1: **Uniform** averaging $\tilde{\rho}_t = \frac{1}{t}\,,\ t \geq 0: \quad \bar{\omega}_T = \frac{1}{T}\sum_{k=0}^{T-1} \omega_t$

Example 2:
Exponential moving averaging (EMA) $\tilde{\rho}_t = 1-\beta < 1\,,\ t \geq 0: \quad \bar{\omega}_T = \frac{1}{1-\beta}\sum_{k=0}^{T-1} \beta^t \omega_t$

Standard Algorithms from Variational Inequality

Method 1: **Averaging**

- Converge even for *"cycling behavior"*.
- Easy to implement. (out of the training loop)
- Can be combined with any method.

General Online averaging: $\bar{\boldsymbol{\omega}}_t = (1 - \tilde{\rho}_t)\bar{\boldsymbol{\omega}}_{t-1} + \tilde{\rho}_t \boldsymbol{\omega}_t \quad \text{where} \quad 0 \le \tilde{\rho}_t \le 1\,.$

<u>Example 1:</u> **Uniform** averaging $\tilde{\rho}_t = \frac{1}{t}\,,\ t \ge 0: \quad \bar{\boldsymbol{\omega}}_T = \frac{1}{T}\sum_{k=0}^{T-1} \boldsymbol{\omega}_t$

<u>Example 2:</u>
Exponential moving averaging (EMA) $\tilde{\rho}_t = 1 - \beta < 1\,,\ t \ge 0\ : \quad \bar{\omega}_T = (1-\beta)\sum_{t=1}^{T} \beta^{T-t}\omega_t + \beta^T \omega_0$

Standard Algorithms from Variational Inequality

Method 1: **Averaging**

- Converge even for *"cycling behavior"*.
- Easy to implement. (out of the training loop)
- Can be combined with any method.

General Online averaging: $\bar{\omega}_t = (1 - \tilde{\rho}_t)\bar{\omega}_{t-1} + \tilde{\rho}_t \omega_t \quad \text{where} \quad 0 \leq \tilde{\rho}_t \leq 1\,.$

Example 1: Uniform averaging $\tilde{\rho}_t = \frac{1}{t}\,,\; t \geq 0: \qquad \bar{\omega}_T = \frac{1}{T}\sum_{k=0}^{T-1} \omega_t$

Example 2:
Exponential moving averaging (EMA) $\tilde{\rho}_t = 1 - \beta < 1\,,\; t \geq 0: \qquad \bar{\omega}_T = \frac{1}{1-\beta}\sum_{k=0}^{T-1} \beta^t \omega_t$

Standard Algorithms from Variational Inequality

Method 1: **Averaging**

- Converge even for *"cycling behavior"*.
- Easy to implement. (out of the training loop)
- Can be combined with any method.

General Online averaging: $\bar{\boldsymbol{\omega}}_t = (1 - \tilde{\rho}_t)\bar{\boldsymbol{\omega}}_{t-1} + \tilde{\rho}_t \boldsymbol{\omega}_t \quad \text{where} \quad 0 \leq \tilde{\rho}_t \leq 1\,.$

<u>**Example 1:**</u> **Uniform averaging** $\tilde{\rho}_t = \frac{1}{t}\,,\; t \geq 0: \qquad \bar{\boldsymbol{\omega}}_T = \frac{1}{T}\sum_{k=0}^{T-1} \boldsymbol{\omega}_t$

<u>Example 2:</u>
Exponential moving averaging (EMA) $\tilde{\rho}_t = 1 - \beta < 1\,,\; t \geq 0\;: \quad \bar{\omega}_T = (1-\beta)\sum_{t=1}^{T} \beta^{T-t}\omega_t + \beta^T \omega_0$

Standard Algorithms from Variational Inequality

Method 1: **Averaging**

Simple Minmax problem: $\min_{\theta \in \mathbb{R}} \max_{\phi \in \mathbb{R}} \ \theta \cdot \phi \quad \Rightarrow \quad (\theta^*, \phi^*) = (0, 0)\,.$

Simultaneous update: $\begin{cases} \theta_{t+1} = \theta_t - \eta \phi_t \\ \phi_{t+1} = \phi_t + \eta \theta_t \end{cases}$, Alternated update: $\begin{cases} \theta_{t+1} = \theta_t - \eta \phi_t \\ \phi_{t+1} = \phi_t + \eta \theta_{t+1} \end{cases}$

Standard Algorithms from Variational Inequality

Method 1: **Averaging**

Simple Minmax problem: $\min_{\theta \in \mathbb{R}} \max_{\phi \in \mathbb{R}} \; \theta \cdot \phi \quad \Longrightarrow \quad (\theta^*, \phi^*) = (0, 0)\,.$

$$\text{Simultaneous update:} \begin{cases} \theta_{t+1} = \theta_t - \eta \phi_t \\ \phi_{t+1} = \phi_t + \eta \theta_t \end{cases}, \qquad \text{Alternated update:} \begin{cases} \theta_{t+1} = \theta_t - \eta \phi_t \\ \phi_{t+1} = \phi_t + \eta \theta_{t+1} \end{cases}$$

$$(\bar\theta_T, \bar\phi_T) := \frac{1}{T} \sum_{k=0}^{T-1} (\theta_t, \phi_t) \to \infty \quad \Big| \quad (\theta_T, \phi_T) \to \infty \quad \Big\| \quad 0 < m \leq \|\theta_T, \phi_T\| \leq M \quad \Big| \quad (\bar\theta_T, \bar\phi_T) \to (0, 0)$$

Standard Algorithms from Variational Inequality

Method 1: **Averaging**

Simultaneous Vs. **Alternating** more developed in
Negative Momentum for Improved Game Dynamics
Gidel, Askari Hemmat, Pezeshki, Lepriol, Huang, Lacoste-Julien and Mitliagkas

Simultaneous update: $\begin{cases} \theta_{t+1} = \theta_t - \eta\phi_t \\ \phi_{t+1} = \phi_t + \eta\theta_t \end{cases}$,

$(\bar{\theta}_T, \bar{\phi}_T) := \frac{1}{T}\sum_{k=0}^{T-1}(\theta_t, \phi_t) \to \infty$ | $(\theta_T, \phi_T) \to \infty$

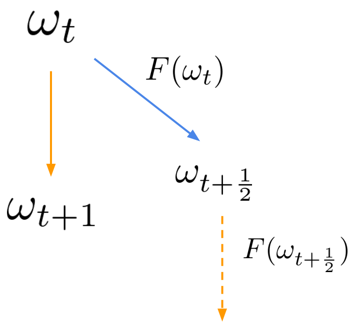
Alternated update: $\begin{cases} \theta_{t+1} = \theta_t - \eta\phi_t \\ \phi_{t+1} = \phi_t + \eta\theta_{t+1} \end{cases}$

$0 < m \leq \|\theta_T, \phi_T\| \leq M$ | $(\bar{\theta}_T, \bar{\phi}_T) \to (0,0)$

Standard Algorithms from Variational Inequality

Method 2: **Extragradient**

- Step 1: $\omega_{t+\frac{1}{2}} = \omega_t - \gamma_t F(\omega_t)$
- Step 2: $\omega_{t+1} = \omega_t - \gamma_t F(\omega_{t+\frac{1}{2}})$



- Standard in the literature.
- Does not require *averaging.*
- *Theoretically* and *empirically* **faster.**

Intuition:

1. *<u>Game prespective:</u> Look one step in the future and anticipate next move of adversary.*
2. *Euler's method: Extrapolation is close to an **implicit** method because* $\omega_{t+1/2} \approx \omega_{t+1}$

$$\omega_{t+1} - \omega_{t+1/2} = O(\gamma_t^2)$$

Standard Algorithms from Variational Inequality

Method 2: **Extragradient**

Intuition: *Extrapolation is close to an* ***implicit*** *method because* $\omega_{t+1/2} \approx \omega_{t+1}$

Implicit step: $\omega_{t+1} = \omega_t - \eta F(\omega_{t+1})$

Unknown:
Require to solve a
non-linear system

Standard Algorithms from Variational Inequality

Method 2: **Extragradient**

Intuition: *Extrapolation is close to an **implicit** method*

$$\min_{\theta \in \mathbb{R}} \max_{\phi \in \mathbb{R}} \quad \theta \cdot \phi \qquad \text{and} \qquad (\theta^*, \phi^*) = (0, 0)\,.$$

$$\text{Implicit:} \begin{cases} \theta_{t+1} = \theta_t - \eta\phi_{t+1} \\ \phi_{t+1} = \phi_t + \eta\theta_{t+1} \end{cases}, \qquad \text{Extrapolation:} \begin{cases} \theta_{t+1} = \theta_t - \eta(\phi_t + \eta\theta_t) \\ \phi_{t+1} = \phi_t + \eta(\theta_t - \eta\phi_t) \end{cases}. \qquad (*)$$

Proposition 2. *The squared norm of the iterates $N_t \overset{def}{=} \theta_t^2 + \phi_t^2$, where the update rule of θ_t and ϕ_t are defined in (*), decreases geometrically for any $\eta < 1$ as,*

$$\text{Implicit:} \;\; N_{t+1} = (1 - \eta^2 + \eta^4 + \mathcal{O}(\eta^6))N_t\,, \qquad \text{Extrapolation:} \;\; N_{t+1} = (1 - \eta^2 + \eta^4)N_t\,.$$

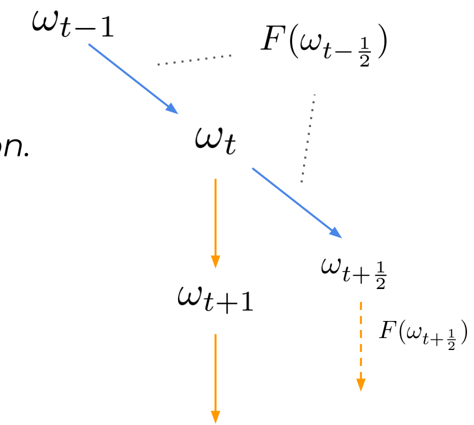
almost the same

Extrapolation from the past: Re-using the gradients

Problem: Extragradient requires to compute **two** gradients at each step.

Solution: **Extrapolation from the past** ← **Re-use** gradient.

- Step 1: $\omega_{t+\frac{1}{2}} = \omega_t - \gamma_t F(\omega_{t-\frac{1}{2}})$ ← ***Re-use*** *from previous iteration.*
- Step 2: $\omega_{t+1} = \omega_t - \gamma_t F(\omega_{t+\frac{1}{2}})$ ← (*same as* ***extragradient***).

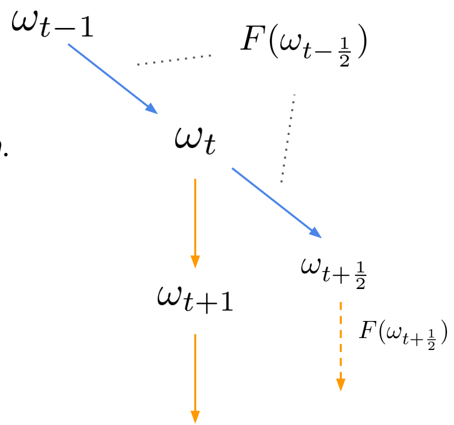


Extrapolation from the past: Re-using the gradients

<u>Problem</u>: Extragradient requires to compute **two** gradients at each step.

Solution: **Extrapolation from the past** ← **Re-use** gradient.

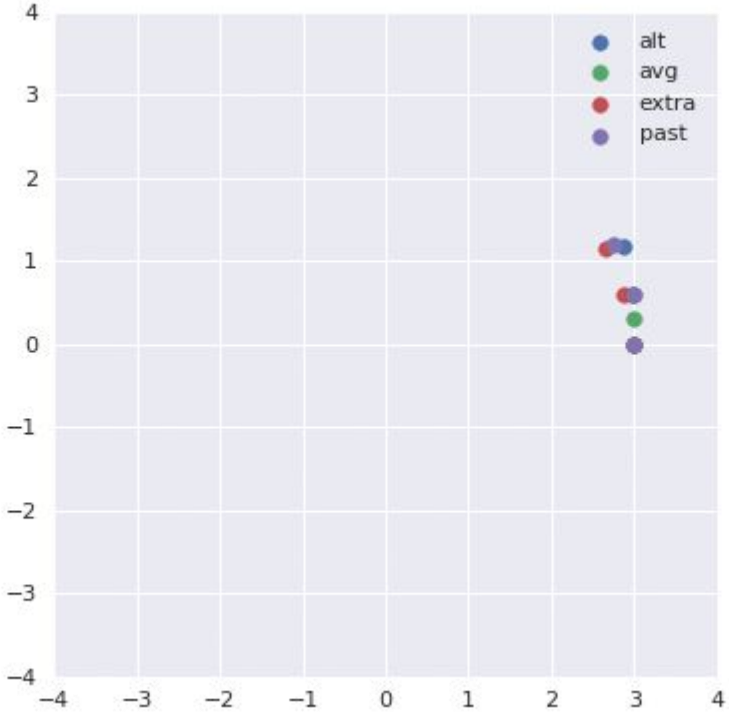
- Step 1: $\omega_{t+\frac{1}{2}} = \omega_t - \gamma_t F(\omega_{t-\frac{1}{2}})$ ← ***Re-use*** *from previous iteration.*
- Step 2: $\omega_{t+1} = \omega_t - \gamma_t F(\omega_{t+\frac{1}{2}})$ ← (*same as* ***extragradient***).



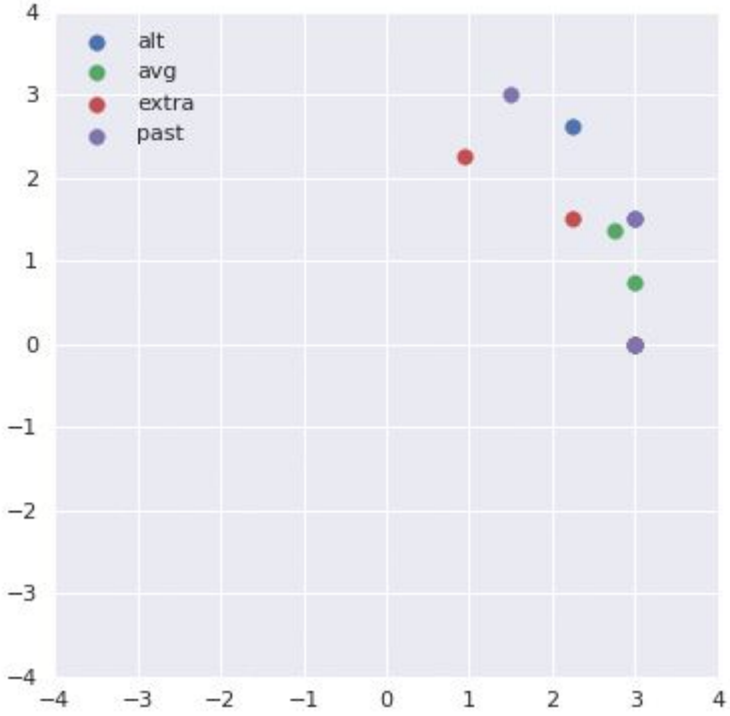
New Method !!!

Related to [Daskalakis et al., 2018]

step-size = 0.2



step-size = 0.5

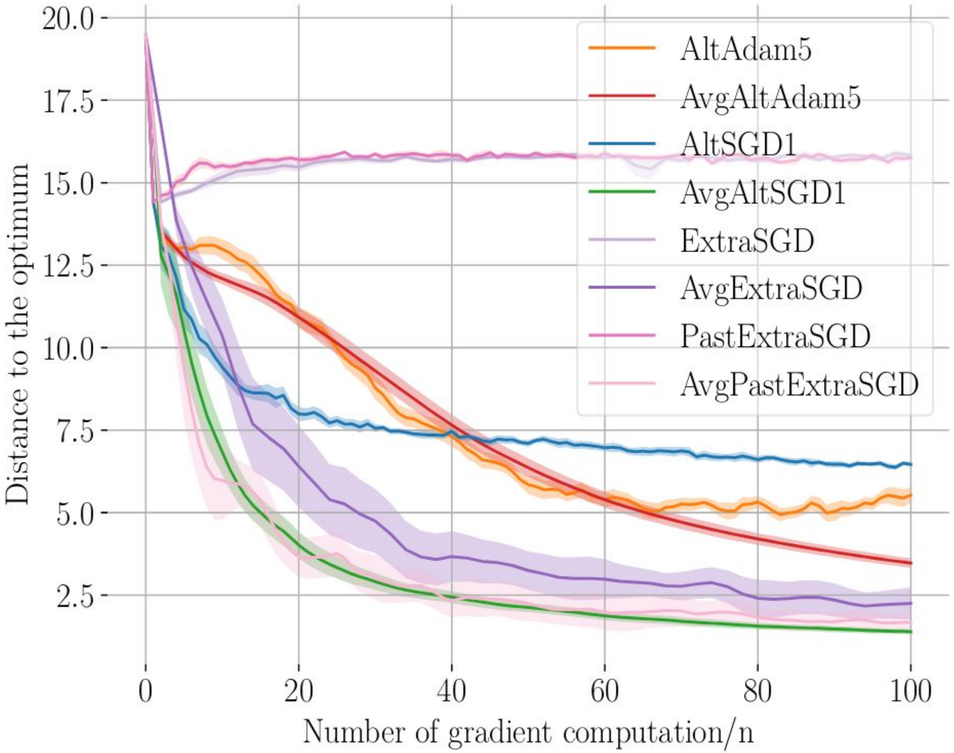


Experimental Results

Experimental Results

Bilinear *Stochastic* Objective: (with constraints)

$$\frac{1}{n}\sum_{i=1}^{n}\left(\boldsymbol{x}^\top \boldsymbol{M}^{(i)}\boldsymbol{y} + \boldsymbol{x}^\top \boldsymbol{a}^{(i)} + \boldsymbol{y}^\top \boldsymbol{b}^{(i)}\right).$$



Algorithm 4 Extra-Adam: proposed Adam with extrapolation step.

input: step-size η, decay rates for moment estimates β_1, β_2, access to the stochastic gradients $\nabla \ell_t(\cdot)$ and to the projection $P_\Omega[\cdot]$ onto the constraint set Ω, initial parameter $\boldsymbol{\omega}_0$, averaging scheme $(\rho_t)_{t \geq 1}$

for $t = 0 \ldots T-1$ **do**

Option 1: Standard extrapolation.

Sample new minibatch and compute stochastic gradient: $g_t \leftarrow \nabla \ell_t(\boldsymbol{\omega}_t)$

Option 2: Extrapolation from the past

Load previously saved stochastic gradient: $g_t = \nabla \ell_{t-1/2}(\boldsymbol{\omega}_{t-1/2})$

Update estimate of first moment for extrapolation: $m_{t-1/2} \leftarrow \beta_1 m_{t-1} + (1-\beta_1) g_t$

Update estimate of second moment for extrapolation: $v_{t-1/2} \leftarrow \beta_2 v_{t-1} + (1-\beta_2) g_t^2$

Correct the bias for the moments: $\hat{m}_{t-1/2} \leftarrow m_{t-1/2}/(1-\beta_1^{2t-1})$, $\hat{v}_{t-1/2} \leftarrow v_{t-1/2}/(1-\beta_2^{2t-1})$

Perform *extrapolation* step from iterate at time t: $\boldsymbol{\omega}_{t-1/2} \leftarrow P_\Omega[\boldsymbol{\omega}_t - \eta \frac{m_{t-1/2}}{\sqrt{v_{t-1/2}}+\epsilon}]$

Extrapolation
(Adam style)

Sample new minibatch and compute stochastic gradient: $g_{t+1/2} \leftarrow \nabla \ell_{t+1/2}(\boldsymbol{\omega}_{t+1/2})$

Update estimate of first moment: $m_t \leftarrow \beta_1 m_{t-1/2} + (1-\beta_1) g_{t+1/2}$

Update estimate of second moment: $v_t \leftarrow \beta_2 v_{t-1/2} + (1-\beta_2) g_{t+1/2}^2$

Compute bias corrected for first and second moment: $\hat{m}_t \leftarrow m_t/(1-\beta_1^{2t})$, $\hat{v}_t \leftarrow v_t/(1-\beta_2^{2t})$

Perform *update* step from the iterate at time t: $\boldsymbol{\omega}_{t+1} \leftarrow P_\Omega[\boldsymbol{\omega}_t - \eta \frac{\hat{m}_t}{\sqrt{\hat{v}_t}+\epsilon}]$

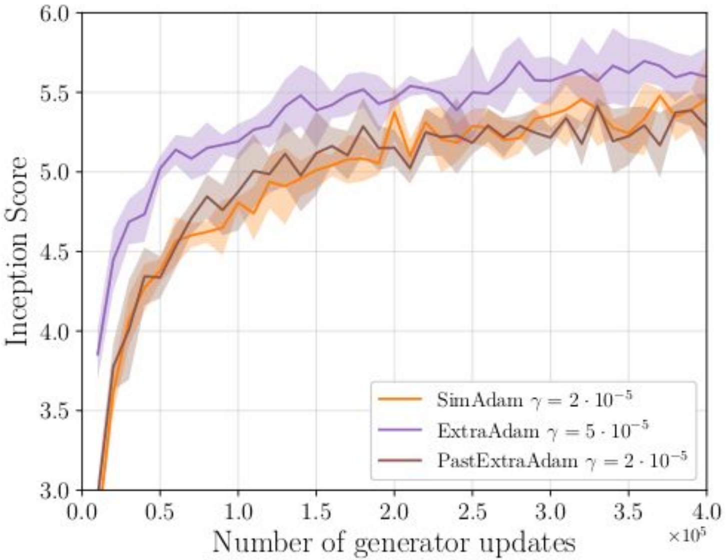
Update
(Adam style)

end for

Output: $\boldsymbol{\omega}_{T-1/2}, \boldsymbol{\omega}_T$ or $\bar{\boldsymbol{\omega}}_T = \sum_{t=0}^{T-1} \rho_{t+1} \boldsymbol{\omega}_{t+1/2} / \sum_{t=0}^{T-1} \rho_{t+1}$ (see (8) for online averaging)

Experimental Results: WGAN on CIFAR10

Inception Score vs
nb of generator updates



Inception Score on CIFAR10

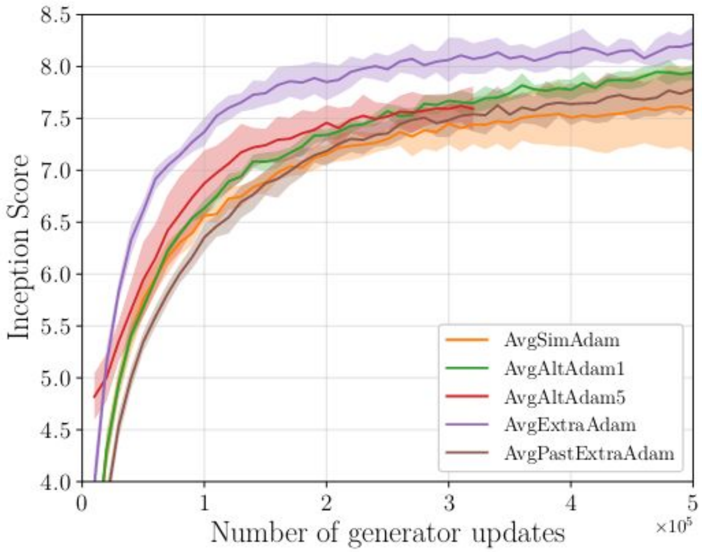
Model	WGAN		
Method	no averaging	uniform avg	EMA
SimAdam	*6.05 ± .12*	5.83 ± .16	6.08 ± .10
AltAdam5	*5.45 ± .08*	5.72 ± .06	5.49 ± .05
ExtraAdam	**6.38 ± .09**	**6.38 ± .20**	**6.37 ± .08**
PastExtraAdam	5.98 ± 0.15	6.07 ± 0.19	6.01 ± 0.11
OptimAdam	5.74 ± 0.10	5.80 ± 0.08	5.78 ± 0.05

Extragradient Methods

Averaging

Experimental Results: WGAN-GP (ResNet) on CIFAR10

Inception Score vs
Number of



Model	WGAN-GP (ResNet)	
Method	no averaging	uniform avg
SimAdam	$7.54 \pm .21$	$7.74 \pm .27$
AltAdam5	$7.20 \pm .06$	$7.67 \pm .15$
ExtraAdam	$7.79 \pm .09$	$\mathbf{8.26} \pm \mathbf{.12}$
PastExtraAdam	$7.71 \pm .12$	$7.84 \pm .18$
OptimAdam	$7.80 \pm .07$	$\mathbf{7.99} \pm \mathbf{.12}$

Extragradient Methods

Averaging

To sum-up

- GAN can be formulated as a **Variational Inequality.**
- Bring *standard methods* from *optimization literature* to the GAN community.
- **Averaging** helps improve the inception score (further evidence by *[Yazici et al. 2018]*).
- **Extrapolation** is **faster** and achieve better convergence.
- Introduce **Extrapolation from the past** a **cheaper** version of *extragradient*.
- We can design better algorithm for GANs inspired from Variational Inequality.

Noise in GANs

Reducing Noise in GAN Training with Variance Reduced Extragradient

Tatjana Chavdarova*[1,2], **Gauthier Gidel***[1], François Fleuret[1,2], Simon Lacoste-Julien[1]

*equal contribution

[1] Mila, Université de Montréal

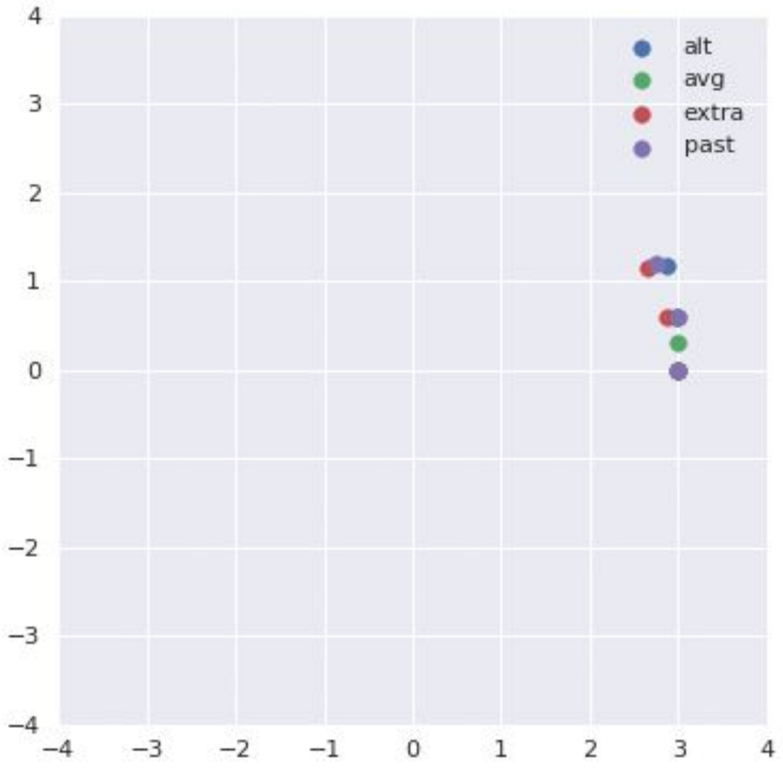
[2] EPFL, IDIAP

Tatjana
Chavdarova

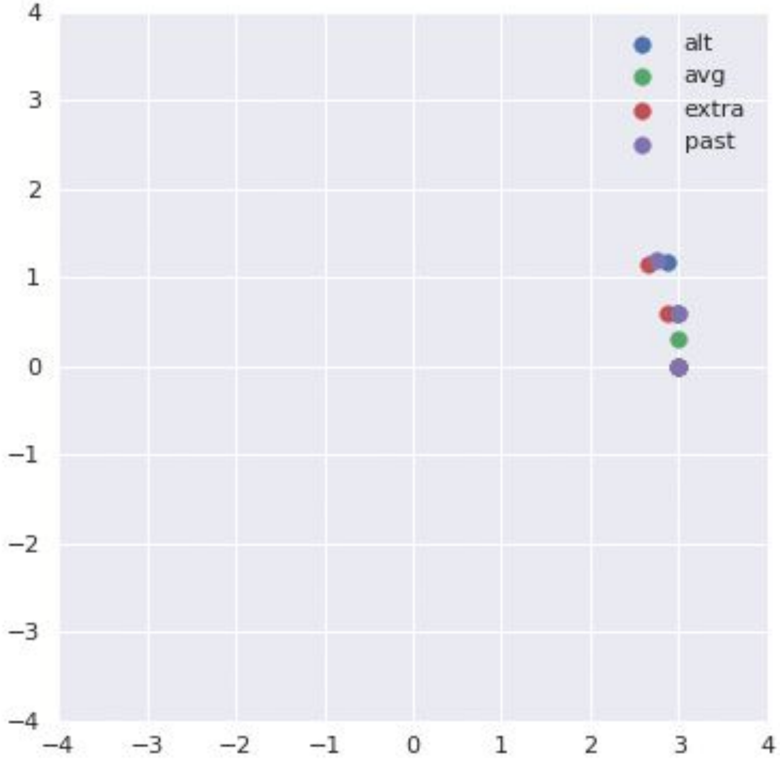
François
Fleuret

Simon
Lacoste-Julien

Reminder: Need for Averaging or/and Extragradient.



Reminder: Need for Averaging or/and Extragradient.



No signal from the average iterate.

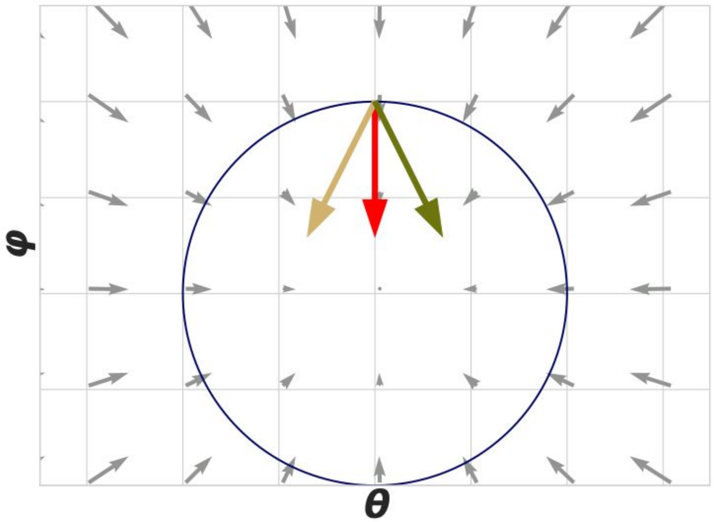
The green sequence **do not stop** at the optimum.

We need **last iterate** convergence.
(Not Convergence of the averaged iterate)

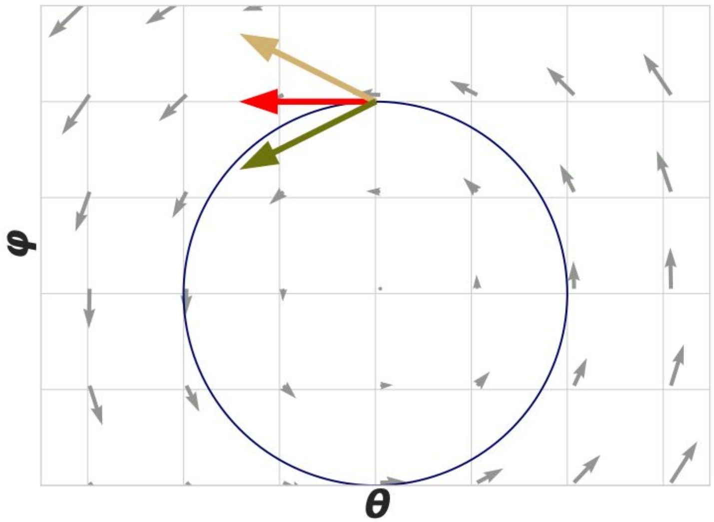
Focus on Extragradient.

Issue: We did not consider **noise.**

Minimization

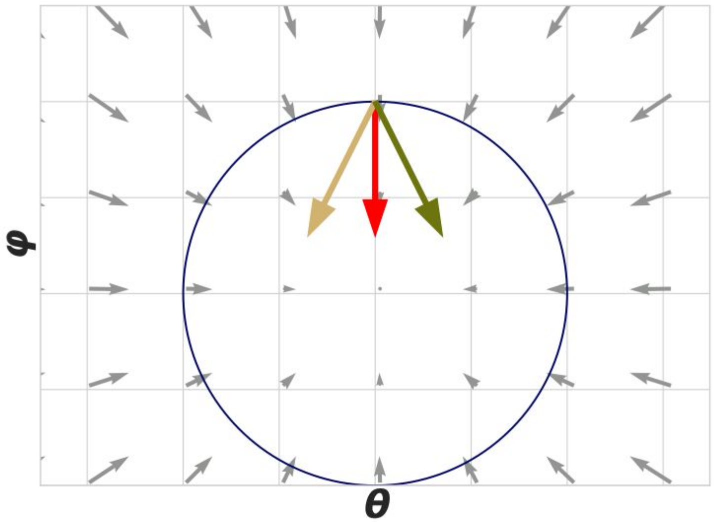


Game

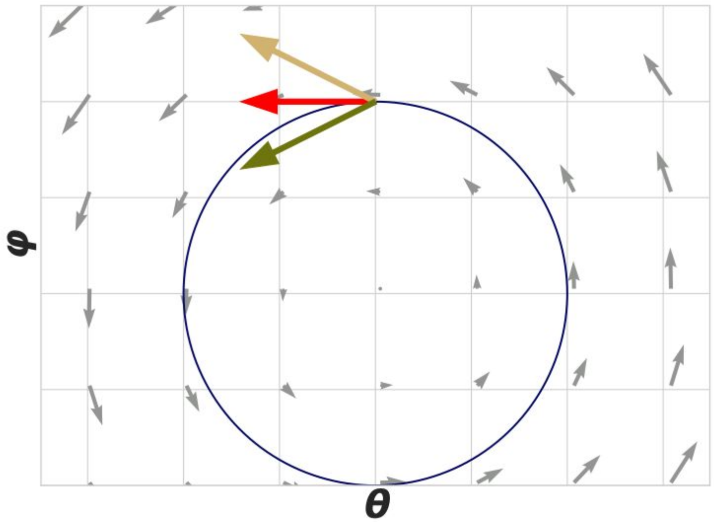


Issue: We did not consider **noise.**

Far from the objective:
“approximately” the right direction



Far from the objective:
Direction with noise can be “bad”.



Standard methods to solve (bilinear) games:

	Gradient method	Extragradient
Batch Method	Diverge to ∞	$\Vert\omega_t - \omega^*\Vert \leq (1-\rho)^t \Vert\omega_0 - \omega^*\Vert$
Stochastic Method	No hope for convergence	**????**

Noise breaks Extragradient.

Theorem 2 (Noise may induce divergence). *There exists a zero-sum stochastic game such that if* $\boldsymbol{\omega}_0 \neq \boldsymbol{\omega}^*$, *then for any step-size* $\eta > 0$, *the iterates* $(\boldsymbol{\omega}_t)$ *computed by the stochastic extragradient method diverge geometrically, i.e., there exists* $\rho > 0$, *such that* $\mathbb{E}[\|\boldsymbol{\omega}_t - \boldsymbol{\omega}^*\|^2] > \|\boldsymbol{\omega}_0 - \boldsymbol{\omega}^*\|^2(1+\rho)^t$.

Noise breaks Extragradient.

Theorem 2 (Noise may induce divergence). *There exists a zero-sum stochastic game such that if $\boldsymbol{\omega}_0 \neq \boldsymbol{\omega}^*$, then for any step-size $\eta > 0$, the iterates $(\boldsymbol{\omega}_t)$ computed by the stochastic extragradient method diverge geometrically, i.e., there exists $\rho > 0$, such that $\mathbb{E}[\|\boldsymbol{\omega}_t - \boldsymbol{\omega}^*\|^2] > \|\boldsymbol{\omega}_0 - \boldsymbol{\omega}^*\|^2(1+\rho)^t$.*

Intuition:

$$\min_{\boldsymbol{\theta} \in \mathbb{R}^d} \max_{\boldsymbol{\varphi} \in \mathbb{R}^d} \frac{1}{n} \sum_{i=1}^{n} \boldsymbol{\theta}^\top \boldsymbol{A}_i \boldsymbol{\varphi}$$

Extragradient Updates:

$$\begin{cases} \boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \eta \boldsymbol{A}_I(\boldsymbol{\varphi}_t + \eta \boldsymbol{A}_J \boldsymbol{\theta}_t) \\ \boldsymbol{\varphi}_{t+1} = \boldsymbol{\varphi}_t + \eta \boldsymbol{A}_I(\boldsymbol{\theta}_t - \eta \boldsymbol{A}_J \boldsymbol{\varphi}_t) \end{cases}$$

Noise breaks Extragradient.

Theorem 2 (Noise may induce divergence). *There exists a zero-sum stochastic game such that if $\boldsymbol{\omega}_0 \neq \boldsymbol{\omega}^*$, then for any step-size $\eta > 0$, the iterates $(\boldsymbol{\omega}_t)$ computed by the stochastic extragradient method diverge geometrically, i.e., there exists $\rho > 0$, such that $\mathbb{E}[\|\boldsymbol{\omega}_t - \boldsymbol{\omega}^*\|^2] > \|\boldsymbol{\omega}_0 - \boldsymbol{\omega}^*\|^2(1+\rho)^t$.*

Intuition:

$$\min_{\boldsymbol{\theta}\in\mathbb{R}^d} \max_{\boldsymbol{\varphi}\in\mathbb{R}^d} \frac{1}{n}\sum_{i=1}^{n} \boldsymbol{\theta}^\top \boldsymbol{A}_i \boldsymbol{\varphi}$$

Extragradient Updates:
(Sample i and j)

$$\begin{cases} \boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \eta \boldsymbol{A}_I(\boldsymbol{\varphi}_t + \eta \boldsymbol{A}_J \boldsymbol{\theta}_t) \\ \boldsymbol{\varphi}_{t+1} = \boldsymbol{\varphi}_t + \eta \boldsymbol{A}_I \underbrace{(\boldsymbol{\theta}_t - \eta \boldsymbol{A}_J \boldsymbol{\varphi}_t)}_{\textbf{Extrapolation part}} \end{cases}$$

$A_i A_j = 0 \Rightarrow$ **No extrapolation**

$\Rightarrow$ **Diverge as GD.**

Reducing noise with Variance reduction methods.

- Idea: take advantage of **the finite sum**.
- Finite sum in ML: Expectation of a **finite** number of sample.
- Generator and discriminator losses can be written as:

$$\mathcal{L}^{(\theta)}(\omega) = \frac{1}{n}\sum_{i=1}^{n} \mathcal{L}_i^{(\theta)}(\omega), \quad \mathcal{L}^{(\varphi)}(\omega) = \frac{1}{n}\sum_{i=1}^{n} \mathcal{L}_i^{(\varphi)}(\omega)$$

SVRG estimate of the gradient.

- Full batch gradient **expensive** but **tractable.**

$$\boldsymbol{d}_i^{(\boldsymbol{\theta})}(\boldsymbol{\omega}) := \left(\nabla \mathcal{L}_i^{(\boldsymbol{\theta})}(\boldsymbol{\omega}) - \nabla \mathcal{L}_i^{(\boldsymbol{\theta})}(\boldsymbol{\omega}^{\mathcal{S}})\right) + \boldsymbol{\mu}_{\boldsymbol{\theta}}^{\mathcal{S}}$$

$$\boldsymbol{d}_i^{(\boldsymbol{\varphi})}(\boldsymbol{\omega}) := \left(\nabla \mathcal{L}_i^{(\boldsymbol{\varphi})}(\boldsymbol{\omega}) - \nabla \mathcal{L}_i^{(\boldsymbol{\varphi})}(\boldsymbol{\omega}^{\mathcal{S}})\right) + \boldsymbol{\mu}_{\boldsymbol{\varphi}}^{\mathcal{S}}.$$

SVRG estimate of the gradient.

Snapshot network

- Full batch gradient **expensive** but **tractable.**

$$\boldsymbol{d}_i^{(\boldsymbol{\theta})}(\boldsymbol{\omega}) := \left(\nabla \mathcal{L}_i^{(\boldsymbol{\theta})}(\boldsymbol{\omega}) - \nabla \mathcal{L}_i^{(\boldsymbol{\theta})}\left(\boldsymbol{\omega}^{\mathcal{S}}\right)\right) + \boldsymbol{\mu}_{\boldsymbol{\theta}}^{\mathcal{S}}$$

$$\boldsymbol{d}_i^{(\boldsymbol{\varphi})}(\boldsymbol{\omega}) := \left(\nabla \mathcal{L}_i^{(\boldsymbol{\varphi})}(\boldsymbol{\omega}) - \nabla \mathcal{L}_i^{(\boldsymbol{\varphi})}\left(\boldsymbol{\omega}^{\mathcal{S}}\right)\right) + \boldsymbol{\mu}_{\boldsymbol{\varphi}}^{\mathcal{S}}.$$

SVRG estimate of the gradient.

- Full batch gradient **expensive** but **tractable.**

Snapshot network

Full gradient at the snapshot network

$$\boldsymbol{d}_i^{(\boldsymbol{\theta})}(\boldsymbol{\omega}) := \left(\nabla \mathcal{L}_i^{(\boldsymbol{\theta})}(\boldsymbol{\omega}) - \nabla \mathcal{L}_i^{(\boldsymbol{\theta})}\left(\boldsymbol{\omega}^{\mathcal{S}}\right)\right) + \boldsymbol{\mu}_{\boldsymbol{\theta}}^{\mathcal{S}}$$

$$\boldsymbol{d}_i^{(\boldsymbol{\varphi})}(\boldsymbol{\omega}) := \left(\nabla \mathcal{L}_i^{(\boldsymbol{\varphi})}(\boldsymbol{\omega}) - \nabla \mathcal{L}_i^{(\boldsymbol{\varphi})}\left(\boldsymbol{\omega}^{\mathcal{S}}\right)\right) + \boldsymbol{\mu}_{\boldsymbol{\varphi}}^{\mathcal{S}}.$$

SVRG estimate of the gradient.

Snapshot network

- Full batch gradient **expensive** but **tractable.**

Full gradient at the snapshot network

$$d_i^{(\theta)}(\omega) := \left(\nabla \mathcal{L}_i^{(\theta)}(\omega) - \nabla \mathcal{L}_i^{(\theta)}\left(\omega^{\mathcal{S}}\right)\right) + \mu_\theta^{\mathcal{S}}$$

$$d_i^{(\varphi)}(\omega) := \left(\nabla \mathcal{L}_i^{(\varphi)}(\omega) - \nabla \mathcal{L}_i^{(\varphi)}\left(\omega^{\mathcal{S}}\right)\right) + \mu_\varphi^{\mathcal{S}}.$$

- **Unbiased** estimates: $\mathbb{E}[d_i^{(\theta)}(\omega)] = \frac{1}{n}\sum_{i=1}^{n} \nabla \mathcal{L}_i^{(\theta)}(\omega) = \nabla \mathcal{L}^{(\theta)}(\omega)$

SVRG estimate of the gradient.

Snapshot network

Full gradient at the snapshot network

- Full batch gradient **expensive** but **tractable.**

$$\boldsymbol{d}_i^{(\boldsymbol{\theta})}(\boldsymbol{\omega}) := \left(\nabla \mathcal{L}_i^{(\boldsymbol{\theta})}(\boldsymbol{\omega}) - \nabla \mathcal{L}_i^{(\boldsymbol{\theta})}\left(\boldsymbol{\omega}^{\mathcal{S}}\right)\right) + \boldsymbol{\mu}_{\boldsymbol{\theta}}^{\mathcal{S}}$$

$$\boldsymbol{d}_i^{(\boldsymbol{\varphi})}(\boldsymbol{\omega}) := \left(\nabla \mathcal{L}_i^{(\boldsymbol{\varphi})}(\boldsymbol{\omega}) - \nabla \mathcal{L}_i^{(\boldsymbol{\varphi})}\left(\boldsymbol{\omega}^{\mathcal{S}}\right)\right) + \boldsymbol{\mu}_{\boldsymbol{\varphi}}^{\mathcal{S}}.$$

- **Unbiased** estimates: $\mathbb{E}[\boldsymbol{d}_i^{(\boldsymbol{\theta})}(\boldsymbol{\omega})] = \frac{1}{n}\sum_{i=1}^{n} \nabla \mathcal{L}_i^{(\boldsymbol{\theta})}(\boldsymbol{\omega}) = \nabla \mathcal{L}^{(\boldsymbol{\theta})}(\boldsymbol{\omega})$
- Compute the snapshot only **once per pass.**

Variance Reduced Extragradient: SVRE

- Combine Extragradient + Variance Reduction for finite sum.

Variance Reduction of Strongly Monotone Games:

Method	Complexity	μ-adaptivity
SVRG	$\ln(1/\epsilon) \times (n + \frac{\bar{L}^2}{\mu^2})$	✗
Acc. SVRG	$\ln(1/\epsilon) \times (n + \sqrt{n}\frac{\bar{L}}{\mu})$	✗
SVRE (This paper)	$\ln(1/\epsilon) \times (n + \frac{\bar{L}}{\mu})$	✓

SVRG and Acc. SVRG are from [Palaniapan and Bach 2016]

Why is this convergence rate not desirable ?

$$\mathbb{E}[Err(\bar{\omega}_T)] \leq O\left(\frac{\sup_{\omega\in\Omega}\|\omega-\omega_0\|\sigma}{\sqrt{T}}\right)$$

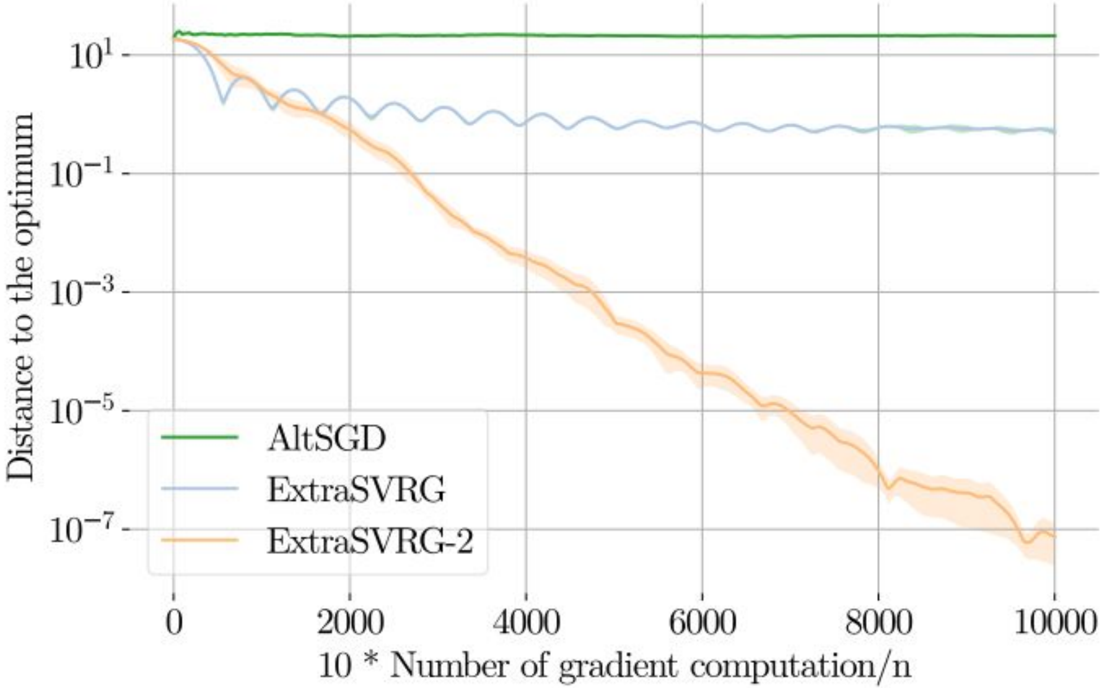
Does not handle **Unconstrained case.**
No restart possible.

Vs.

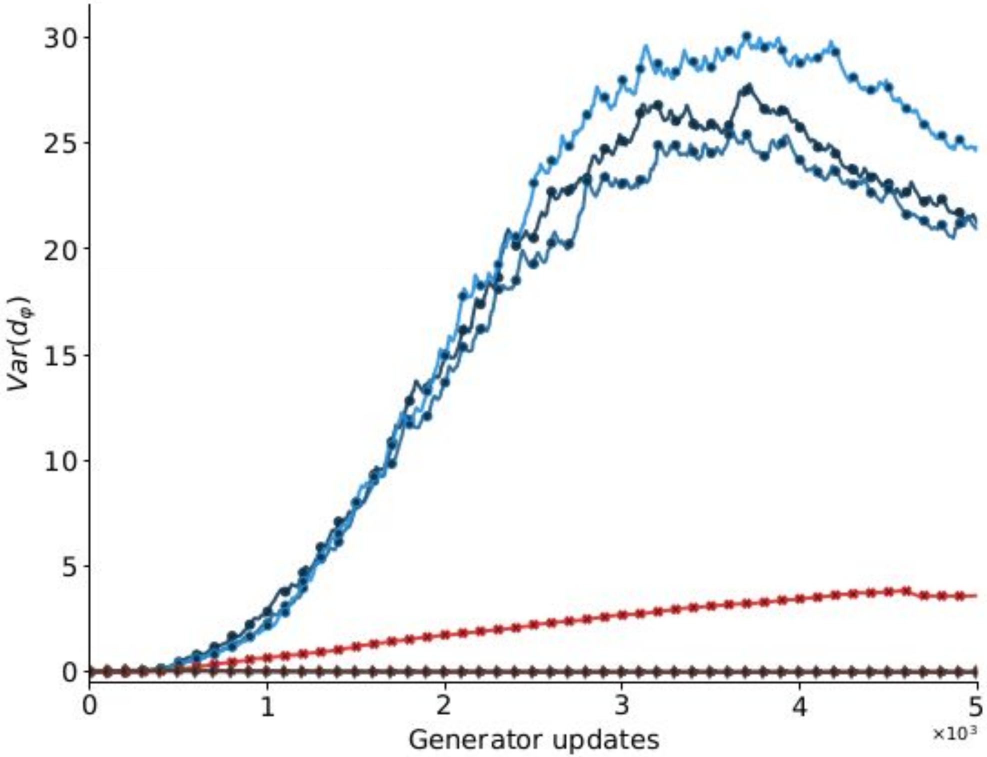
$$\mathbb{E}[Err(\bar{\omega}_T)] \leq O\left(\frac{\|\omega^*-\omega_0\|\sigma}{\sqrt{T}}\right)$$

Does handle **Unconstrained case.**
Restart possible.

SVRE on bilinear Game:
(Exact example where stochastic extragradient breaks)



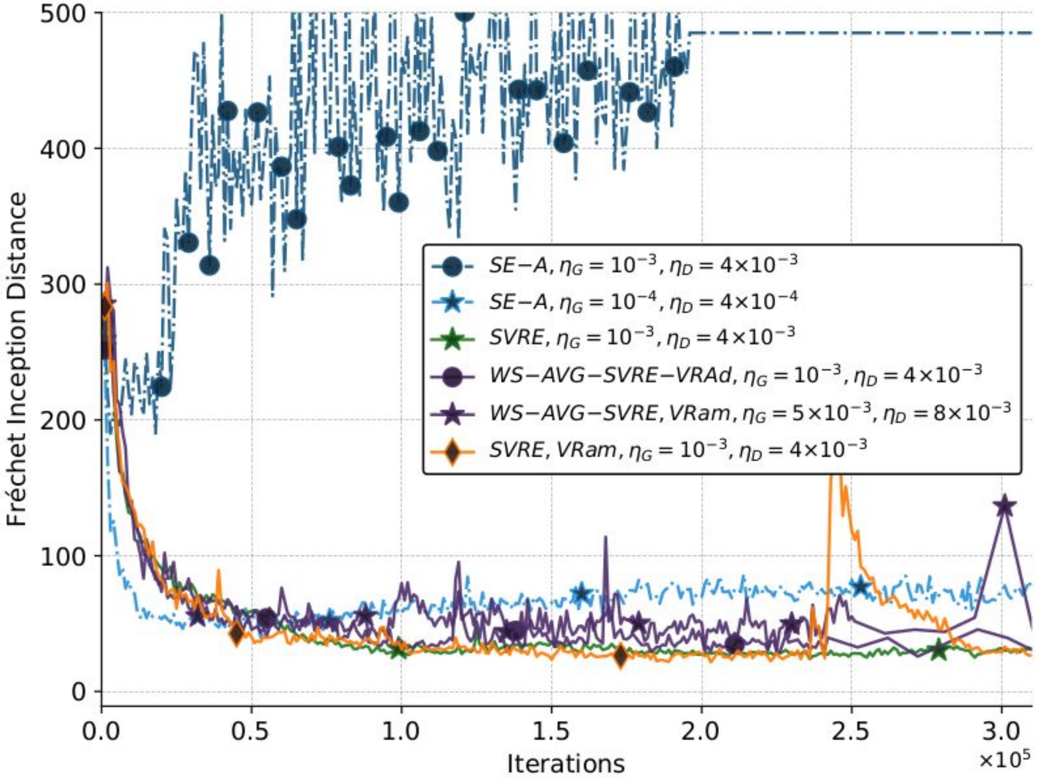
First point, SVRE effectively reduces the variance:



Blue: Stochastic Extragradient

Brown: SVRE.

Second point SVRE allows larger step-sizes: (SVHN)



SE: Stochastic Extragradient.

SVRE: Variance Reduced Extragradient.

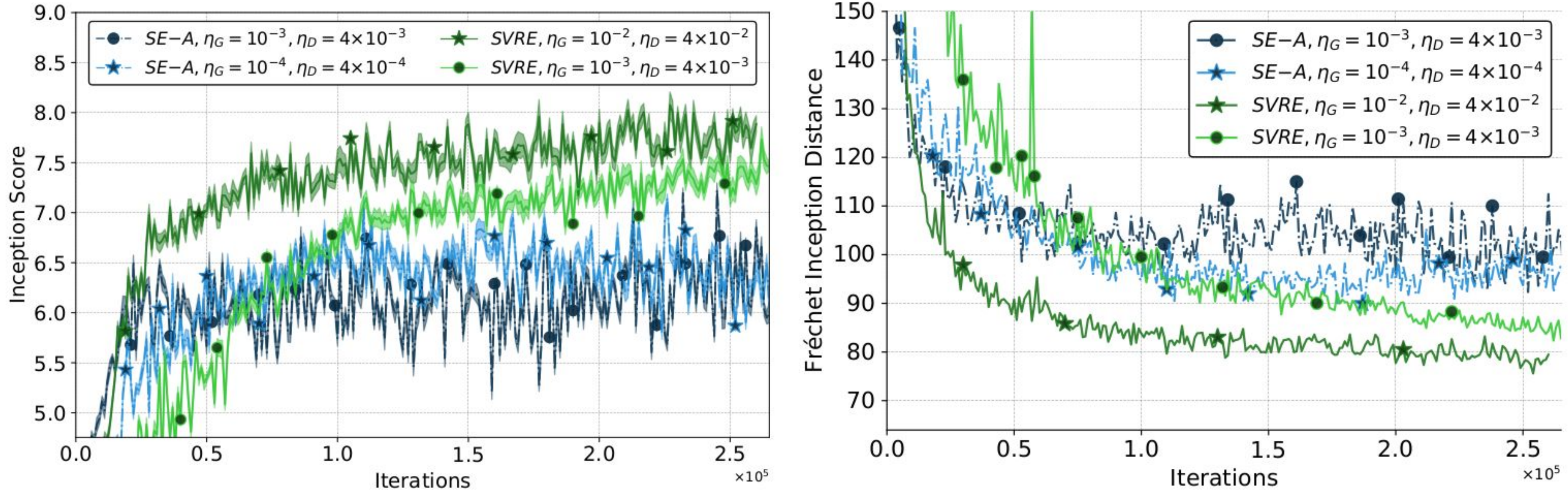
-A: Adam

WS: Warm Start.

AVG: Average.

-VRAd (VRam): variant of Adam for SVRE.

Second point SVRE allows larger step-sizes: (ImageNet)



(a) IS (higher is better) (b) FID (lower is better)

To sum-up

- **Noise** may be an issue in GANs.
- Proposed to combine VR + Extragradient to tackle **both** game and noise aspects.
- Unlike in single-objective minimization, we observed that **variance reduction could improve the performance** of deep learning models for GAN training.
- highlights the difference between **game** optimization and **standard minimization.**